

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA APLIKOVANÉ INFORMATIKY

Internetové rozhraní pro zakázkovou výrobu elektrických kytar
Web Interface for Custom Manufacturing of Electric Guitars

Student: Tomáš Tomis
Vedoucí bakalářské práce: Ing. Martin Pochyla, Ph.D.

Ostrava 2014

Zadání bakalářské práce

Student:

Tomáš Tomis

Studijní program:

B6209 Systémové inženýrství a informatika

Studijní obor:

6209R001 Aplikovaná informatika

Téma:

Internetové rozhraní pro zakázkovou výrobu elektrických kytar
Web Interface for Custom Manufacturing of Electric Guitars

Zásady pro vypracování:

1. Úvod
2. Teoretická východiska použití HTML 5.0
3. Analýza současného stavu webů konkurenčních výrobců kytar
4. Návrh internetového rozhraní
5. Řešení a implementace rozhraní
6. Závěr

Seznam použité literatury

Seznam zkratk

Prohlášení o využití výsledků bakalářské práce

Seznam příloh

Přílohy

Seznam doporučené odborné literatury:

PILGRIM, M., L. LAZARIS and E. WEYL. *HTML5: up and running*. Sebastopol. CA: O'Reilly, 2010. ISBN 05-968-0602-7.

LUBBERS, P., B. ALBERS a F. SALIM. *HTML5 - Programujeme moderní webové aplikace*. Přeložil Ondřej GIBL. Brno: Computer Press, 2011. ISBN 978-80-251-3539-6.

SARRIS, Simon. *HTML5 Unleashed*. Indianapolis: Sams Publishing, 2013. ISBN 978-06-723-3627-0.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Martin Pochyla, Ph.D.**

Datum zadání: 22.11.2013

Datum odevzdání: 09.05.2014



Ing. Petr Rozehnal, Ph.D.
vedoucí katedry

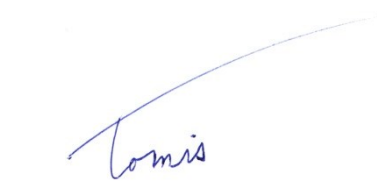


prof. Dr. Ing. Dana Dluhošová
děkanka fakulty

Zde bych chtěl velmi poděkovat Ing. Martinu Pochylovi Ph.D., za ukázkovou spolupráci, dobré rady a profesionální přístup při vedení mé bakalářské práce.

„Prohlašuji, že jsem celou práci, včetně všech příloh, vypracoval samostatně“.

V Ostravě dne 7. května 2014



Tomáš Tomis

Adresa trvalého pobytu studenta:

Na Sedlácích 1011

739 34 Šenov

Obsah

1.	Úvod.....	6
2.	Teoretická východiska použití HTML5	7
2.1.	Historie jazyka HTML.....	8
2.2.	Jazyk HTML5	9
2.3.	Struktura HTML5	10
2.3.1.	Header	10
2.3.2.	Section.....	10
2.3.3.	Article.....	10
2.3.4.	Nav	11
2.3.5.	Aside	11
2.3.6.	Hgroup.....	11
2.3.7.	Footer	11
2.3.8.	Time	11
2.3.9.	Mark	11
2.4.	Podpora HTML5.....	13
2.5.	Srovnání HTML4 a HTML5	15
2.6.	Nejpoužívanější Pluginy HTML5	16
2.6.1.	Kontaktní formulář AJAX.....	16
2.6.2.	Prezentační plugin CCSlider	16
2.6.3.	Hudební přehrávač MusicBox.....	16
2.6.4.	Easy Video Background.....	16
2.6.5.	HTML5 Video & YouTube Background	17
2.7.	CSS	17
2.8.	Historie CSS	17
2.9.	CSS3	18
2.9.1.	Selektory	18

2.9.2.	Barvy	19
2.9.3.	Zaoblení rohů	20
2.9.4.	Stínování	21
2.10.	Prvek Canvas	22
2.10.1.	Historie Canvas	22
2.10.2.	Vytvoření Canvas	23
2.10.3.	Kreslení pomocí Canvas	23
3.	Analýza současného stavu webů konkurenčních výrobců kytar	24
3.1.	Firma Warmoth	25
3.2.	Firma Usa Custom Guitars	27
3.3.	Firma Fender	28
3.4.	Moniker Guitars	30
3.5.	Analýza stavu firmy	31
4.	Návrh internetového rozhraní	32
4.1.	Layout stránky	32
4.1.1.	Záhlaví a zápatí	34
4.1.2.	Canvas	34
4.1.3.	Navigace	34
4.1.4.	Tlačítka	35
4.1.5.	Cena	35
5.	Řešení a implementace rozhraní	35
5.1.	Realizace layoutu	35
5.1.1.	Body	35
5.1.2.	Header	36
5.1.3.	Footer	36
5.1.4.	Canvas	36
5.1.5.	Nav	37

5.1.6.	Select	38
5.1.7.	Button	39
5.1.8.	Box a FinalPrice	40
5.2.	Realizace vykreslení kytary	41
5.2.1.	Tvorba grafiky	41
5.2.2.	Logika vykreslování	42
5.2.3.	JavaScript pro vykreslení	43
5.2.4.	Podmínky vykreslení	44
5.2.5.	Počítání ceny	46
5.2.6.	Vyčištění zvolených možností	47
5.2.7.	Tisk kytary	49
5.2.8.	Uložení obrázku kytary	50
5.2.9.	Objednávka pomocí JotForm	51
5.3.	Údržba a přidávání nových prvků	53
5.3.1.	Rozšíření stávajících možností	53
5.3.2.	Přidání nové možnosti	54
6.	Závěr	56
	Seznam použité literatury	58
	Tištěné knihy	58
	Internetové zdroje	58
	Seznam zkratk	59

1. Úvod

Internet je světovým fenoménem a v posledních letech se stává stále důležitější součástí našich životů. Používáme ho dnes a denně, ať už z pracovních důvodů nebo kvůli zábavě.

Historie internetu a globálně propojeného webu se datuje již od roku 1982. Internet začal vznikat postupným propojováním menších sítí pomocí systému protokolů, na kterých funguje více méně až do dnešních dnů. Jedna z hlavních služeb, kterou internet může poskytovat, známá jako World Wide Web, se tedy stala mocnou zbraní a to jak na poli komunikace, tak na poli obchodu a celosvětového kulturního potažmo sociálního rozvoje.

Je také velmi důležitým obchodním a marketingovým médiem a každá menší i větší firma by měla využít jeho globálního rozsahu ke zlepšení svého podnikání, prodeje a zisků. Vysoká kvalita webové prezentace produktů je v posledních letech nutností. Pro dnešní hlavně mladší generaci je nakupování na internetu a přístup k informacím každodenní záležitostí, což jednoznačně naznačuje, že internet se může stát vysoce efektivním propagačním médiem. I s relativně malými náklady lze dnes propagovat na sociálních sítích obrovské množství produktů. Proto také někteří výrobci přechází od placené reklamní činnosti, popř. z placené internetové reklamy na internetovou reklamu zaměřenou výhradně na sociální síť a virální marketing. Reklama byla přenesena z reklamních agentur na spotřebitele samotné. Velké množství spotřebitelů běžně hodnotí firmy na sociálních sítích, často také sdílí odkaz na webové stránky firem nebo na jejich profil na sociálních sítích.

Grafice webu je dnes přikládána důležitá role, grafika totiž v prvních sekundách návštěvy rozhodne, zda návštěvník potažmo potenciální zákazník na stránce zůstane nebo ne. Moderní trendy tvorby webu se vracejí zpět ke kořenům. Novodobé weby jsou psány stroze za použití jednoduchého rozvržení a písma. Upouští se od přehnaného množství obrázků a zbytečných „kudrlinek“. Dalo by se říci, že se řídí heslem: „V jednoduchosti je síla“. Proto jsou dnes v oblibě jednostránkové scrollovací weby, které mají důrazně definované nejpodstatnější elementy a snaží se, aby od prvního momentu bylo uživateli vše jasné. Jednoduché grafické prvky se často dynamicky pohybují nebo mění pomocí JavaScriptu. Proto vzniká spousta strohých vstupních stránek, které například nabízí nějaký produkt. Nemusí se ani jednat o hmotný produkt, může to být třeba služba. Celý popis produktu je stroze a výstižně napsán v několika větách. A v podstatě jedním klikem lze produkt, jednoduše a pohodlně koupit. Celou řadu grafických efektů, které se kdysi musely složitě vytvářet na rastrových obrázcích pomocí grafických programů, je dnes možno vytvořit

několika řádky kódu CSS. Další stěžejní záležitostí je přizpůsobení grafiky webu přístupu z mobilních zařízení. Jedná se o velmi rozsáhlou problematiku a existuje mnoho postupů jak tyto záležitosti řešit. Většinou se stránka od počátku tvoří tak, aby se dobře zobrazovala na tabletech, chytrých telefonech atd., nebo se vytvoří speciální verze stránky, která se bude zobrazovat jen uživatelům, kteří přistupují z mobilních zařízení.

Z toho vyplývá nutnost firem mít nejen kvalitní, dobře fungující webovou stránku a dobře zpracovaný web po grafické stránce, ale také mít vytvořen firemní profil na nejnavštěvovanějších sociálních sítích a neustále udržovat kontakt se stálými i potenciálními zákazníky. Lidé musí mít neustále pocit, že se ve firmě něco děje, zlepšují se služby a že firma dělá vše pro spokojenost zákazníků. U specializovaných firem je vhodné, aby měly vytvořeny potřebné webové aplikace související s jejich podnikáním nebo s přímou komunikací se zákazníky. Proto vzniká obrovské množství dynamických webových stránek a webových aplikací ovládaných JavaScriptem po případě aplikací vytvořených ve Flash Playeru podobně. Velké množství zákazníků chce svůj vysněný produkt předem vidět a mít možnost volby a úprav produktu, dle svého vkusu a osobních preferencí. Náročnější zákazníci se dnes často odvrací od velkovýroby a jsou ochotni si připlatit nemalé sumy za zboží, které je jim tzv. „ušito na míru“. Záležitostí týkajícím se virtuální výroby produktu, který je přesným obrazem zákaznickových představ se budeme věnovat v následujících stránkách.

Cílem práce je vytvořit webovou aplikaci pro firmu zabývající se výrobou vysoce specifických, zakázkových elektrických kytar, které budou přesně přizpůsobeny požadavkům zákazníka. Zákazník si sám pomocí této aplikace bude schopen svůj vysněný nástroj do detailu nadefinovat a udělat si představu o vzhledu a výsledné ceně nástroje. Firma počítá s tím, že cílovou skupinou nebudou začínající kytaristé, ale zkušení hudebníci z branže, kteří vědí, co chtějí a hledají profesionální nástroj na desítky let. Bude použito technologií HTML5, CSS3 a jazyku JavaScript k plnému využití moderních možností, které tyto technologie nabízí.

2. Teoretická východiska použití HTML5

V této části se budeme zabývat jazykem pro psaní webových stránek HTML verze 5.0. Budeme se věnovat jeho formě, základům jeho používání, syntaxi a v nemalé části taky historii tohoto jazyka. Objeví se zde také popis a informace o dalších technologiích, které se

týkají tvorby webu jako například kaskádové styly CSS a nebo také informace o JavaScriptu a jeho knihovnách, které budou v práci taktéž využity.

2.1.Historie jazyka HTML

Jazyk HTML byl vytvořen Britem jménem Timothy Berners-Lee v roce 1991. Měl sloužit pro rychlou celosvětovou komunikaci vědcům, kteří zkoumali fyziku vysokých energií. Mohli tak sdílet své výsledky a poznatky podstatně rychleji než doposud. Celý tento projekt vznikl ve Švýcarsku v CERNu (Evropská organizace pro jaderný výzkum). První verze HTML byla známá jako HTML 0.9 a umožňovala dělení textu do několika úrovní, různá zvýraznění textu a také uměla přidat odkazy a obrázky. (Kosek, 1998)

Původní verze měla být taková, že prohlížeč program bude zároveň obsahovat i HTML editor. Takže by pisatel WWW stránek nemusel vůbec znát HTML jazyk. Nicméně implementace editoru do prohlížeče byla nakonec považována za velmi obtížnou, tak se od tohoto nápadu upustilo. Proto je až do dnešních dní znalost HTML při psaní webu nezbytná. (Kosek, 1998)

Požadavky na složitost webů se zvyšovaly, a proto bylo třeba HTML zlepšovat. Začalo vznikat velké množství prohlížečů od různých firem. Vývoj dospěl až k další verzi HTML 2.0, ve které Berners-Lee zachoval, popř. vylepšil prvky z původní verze, ale zároveň přidal i některé nové možnosti použití formulářů. Další rozšíření obsahují tvorbu matematických vzorců, tabulek a také mnohem lepší práci s textem a jeho vzhledem, obtékání obrázků textem, stylování dokumentů a podobně. Dohromady se všem těmto úpravám říkalo HTML+, ovšem v roce 1995 pracovník firmy Hawlett-Packard Dave Raggett vytvořil a deklaroval jednu společnou formulaci těchto změn známou jako HTML 3.0 (Kosek, 1998).

Problémem však bylo, že nejnovější verze prohlížečů Mosaic a Netscape nepodporovaly všechny prvky HTML 3.0. Jediný prohlížeč, který je podporoval, byl prohlížeč Arena, který byl ovšem spíše experimentem a podporovaly ho jen operační systémy Unix. (Kosek, 1998)

Konsorciu W3C, které se stará o vývoj standardů webu bylo tehdy jasné, že HTML 3.0 je příliš pokrokové a bude ještě nějakou dobu trvat, než bude vyvinut prohlížeč schopný podporovat všechny jeho možnosti. Proto se dohodli na kompromisu a vytvořili HTML 3.2, což bylo v podstatě HTML 2.0, které obsahovalo jen některá rozšíření z HTML 3.0. (Kosek, 2013)

Od roku 1996 se začaly používat kaskádové styly, což vlastně znamená převedení veškerého stylování webu do jednoho externího CSS souboru. Následně v roce 1997 vznikl HTML 4.0 a krátce poté i úplně nový standard XML (eXtensive Markup Language), který sloužil především pro ukládání a výměnu dat. Poté následoval i jazyk XHTML založený na XML. Po čase se ukázalo, že XHTML je spíše záležitostí trendu a technické výhody nepřináší. Začalo se pracovat na jeho nástupci XHTML 2.0, ale problém byl v tom, že výrobci prohlížečů jej odmítali využívat, takže byl jeho vývoj nakonec úplně zastaven. (Kosek, 2013)

Vše vedlo k vylepšení HTML4, kterým se stalo HTML5. HTML5 přidává obrovskou škálu užitečných funkcí a zároveň umožňuje používat i syntaxi XML potažmo XHTML. Tento standart je však stále otevřený a i když se již používá, pořád se na něm pracuje. Nicméně nemalá část HTML5 je v dnešní době již podporována většinou prohlížečů. (Kosek, 2013)

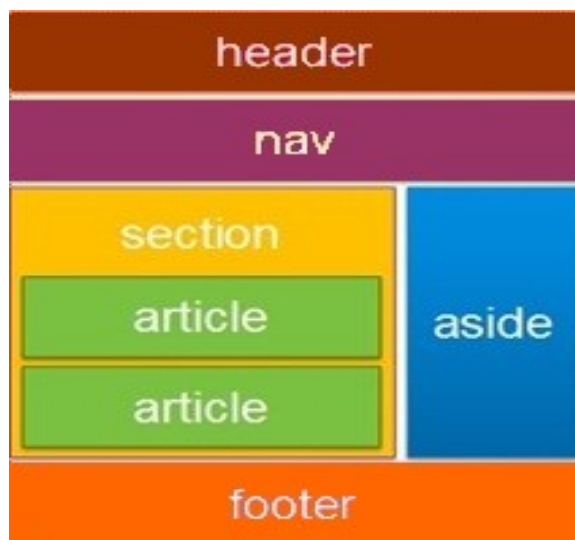
2.2.Jazyk HTML5

HTML neboli Hyper Text Markup Language je jazyk, který se používá pro psaní webových stránek. Je hojně rozšířený a jeho poslední verze 5.0 přináší velké množství nových prvků. Je dominantním jazykem pro zobrazování dat a obsahu na internetu potažmo WWW. Verzi HTML5 nesmíme chápat jako nějakou přeměnu pravidel a zásad z předchozí verze 4.0. Klasické syntaxe zde fungují stejně jako v předchozích verzích nicméně jsou obohaceny o nové prvky, nové možnosti a zlepšení starých prvků.

HTML5 především obsahuje předefinování stávajících značkovacích elementů a nové prvky, které umožňují webovým designérům být výraznější v sémantice jejich značek. Není důvod proč dávat na stránky pouze odstavce, když můžeme mít články, oddíly, záhlaví, zápatí a další. Výraz HTML5 byl navíc používán jako odkaz na řadu dalších nových technologií a rozhraní API. Mezi některé z nich patří kreslení s canvas prvkem, což je plátno, offline uložení, nové video, audio prvky, funkce drag-and-drop přesouvání pomocí myši, Mikrodata, vložené fonty a další. (Goldstein, 2011)

2.3. Struktura HTML5

V této části se budeme nejprve věnovat novým sémantickým prvkům HTML5. Na obrázku můžeme vidět výčet a rozložení prvků HTML5.



Obrázek 2.1. prvky HTML5 zdroj: (vlastní obrázek)

2.3.1. Header

Je to jeden ze základních prvků. Bylo by dobré, kdyby zahrnoval název stránky nebo nějaké logo či slogan. Často se zde také přidává prvek k navigaci webu. Po záhlaví se obsah dělí na tři sloupce. Záhlaví je možno používat různých typů pro různý obsah. (Goldstein, 2011)

2.3.2. Section

Následují bloky section. Které, ač k tomu svádí, není radno používat k vizuálnímu rozdělení stránky, ale pouze k jejímu logickému rozdělení. Jako bychom měli například sekci sport a inzeráty v novinách. K vlastnímu vizuálnímu rozdělení do sloupců budou postačovat klasické tagy div, popř. můžeme využít i vlastnosti prvku article. (Goldstein, 2011)

2.3.3. Article

Ideální prvek pro vkládání článků, jak už sám název napovídá. Do divů, které jsme si vytvořili v sekci. Měl by obsahovat prvky, které budou nezávisle šířitelné, a také bude možnost je znovu použít. Takže jak bylo výše zmíněno pro vložení článků v podobě dokumentu, aplikace, interaktivního ovládacího prvku atd. (Pilgrim, 2010)

2.3.4. Nav

Prvek pro navigaci. Odkazujeme se pomocí ní na jiné stránky nebo na vnitřní prvky stránky samotné. Všechny odkazy nemusí být součástí nav elementu. Používá se hlavně pro důležité odkazy a pro hlavní navigační bloky. Odkazy na podobné weby nebo spřízněné weby často bývají i v prvku zápatí. (Pilgrim, 2010)

2.3.5. Aside

Používá se pro vytváření postranních elementů, které bývají velmi často nazývány sidebary. Používá se pro prvky jako postranní lišty, pro nav elementy nebo také třeba pro různé reklamní a propagační části nebo pro jakýkoliv obsah oddělený od hlavní strany. (Pilgrim, 2010)

2.3.6. Hgroup

Prvek, který slouží pro tvorbu nadpisů v sekci. Obsahuje soubor nadpisů klasicky dělených od h1 až po h6 podle potřeby. Což může sloužit k vytvoření úrovní nadpisů, podnadpisů atd. (Pilgrim, 2010)

2.3.7. Footer

Slouží pro tvorbu zápatí. Tento prvek se vyskytuje ve spodní části stránky. Většinou slouží k deklaraci autorských práv, informací o autorovi nebo k výpisu odkazů. Nemusí se vždy dávat na konec sekce, ovšem ve většině případů tam bývá umístěn. Může ovšem také sám obsahovat celé sekce. (Pilgrim, 2010)

2.3.8. Time

Typický prvek času v klasickém 24 hodinovém formátu a přesné datum podle gregoriánského kalendáře, s volitelným nastavením časového posunu a času samotného. (Pilgrim, 2010)

2.3.9. Mark

Tento element představuje označení pole textu souvislým zvýrazněním. Používá se z důvodu zvýraznění nebo reference. (Pilgrim, 2010)

Nyní si ukážeme kus kódu v HTML5 jazyce. Jeho struktura se dá dělit do několika částí. Ovšem ani ty nejrozsáhlejší a nejsložitější stánky nemohou využít všech možností, které se ve verzi 5.0 skýtají. Proto si ukážeme jen jednoduchou verzi kódu, který mohou stránky obsahovat, pro praktické vysvětlení některých výše zmíněných prvků.

```
<body>

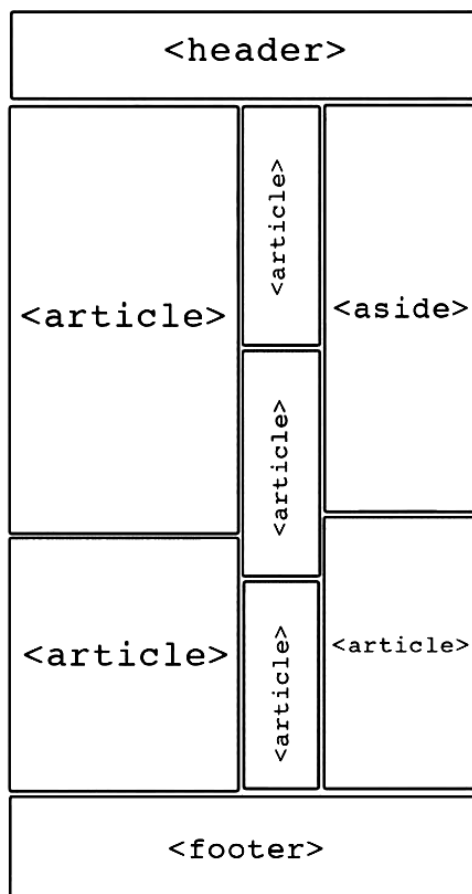
  <header>
<nav></nav>
</header>

<div id="hlavni">
  <div id="primarni">
    <article></article>
    :
  </div>
  <div id="sekundarni">
    <article></article>
    :
  </div>
  <div id="terciarni">
    <aside>
      <article></article>
      :
    </aside>
    <article></article>
  </div>
</div><!-- #main -->

<footer>
  <section id="autor">
    <section></section>
  </section>
  <section id="copyright"></section>
</footer>

<script src="js/scripts.js"></script>
</body>
```

Na dalším obrázku můžeme vidět, jak bude vypadat layout výše zmíněného kódu. Hned na první pohled jsou patrné jasné rozdíly oproti HTML4.



Obrázek 2.2. Zobrazení layoutu zdroj: (Goldstein, 2011)

2.4.Podpora HTML5

HTML5 je dnes buď z části, nebo téměř zcela podporováno velkým množstvím internetových prohlížečů pro osobní počítače, ale také pro mobilní zařízení. Pro přehlednost si ukážeme tabulku, ve které můžeme vidět nejpoužívanější prohlížeče jako Mozilla Firefox, Google Chrome, Safari a jiné. Dále zde vidíme prvky, které jednotlivé prohlížeče podporují, nepodporují nebo částečně podporují.

2.5.Srovnání HTML4 a HTML5

Největší rozdíly se týkají prvků výše zmíněných prvků stránky jako header, footer, nav, article atd. Upouští se od takzvaného „předivování“, což je přehnané používání prvku div, kterým byly starší verze HTML typické. Také je rozdíl v obrovském zjednodušení doctype, což je prvek, který musí být uveden na začátku každé stránky. U HTML5 stačí napsat pouze `<!DOCTYPE html>`. HTML5 jako takové je ovšem na HTML4 postaveno. Takže všechny klasické věci fungují i ve verzi 5. Velkým rozdílem je ovšem i podpora prohlížečů. Jak už bylo zmíněno výše, tím že HTML5 přidává množství nových audiovizuálních prvků, stává se nekompatibilním se staršími verzemi prohlížečů. Až do dnešní doby nebyl vyvinut takový prohlížeč, který by podporoval všechny možnosti, které HTML5 nabízí. Opět si ukážeme rozdíly i na kusu kódu.

HTML4

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
  <meta http-equiv="content-type" content="text/html;
charset=utf-8">
  <title>Basic</title>
</head>
<body>
  <div id="header">
    <div id="navigation">
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">FAQ</a></li>
        <li><a href="#">Contact Us</a></li>
      </ul>
    </div>
  </div>
  <div id="article">
    <h1>Welcome</h1>
    <p>Welcome to our site</p>
  </div>
  <div id="sidebar">
    <p>Information that goes on a sidebar</p>
  </div>
  <div id="footer">Copyright 2010</div>
</body>
</html>
```

HTML5

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Basic</title>
</head>
<body>
  <header>
    <nav>
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">FAQ</a></li>
        <li><a href="#">Contact Us</a></li>
      </ul>
    </nav>
  </header>
  <article>
    <h1>Welcome</h1>
    <p>Welcome to our site</p>
  </article>
  <aside>
    <p>Information that goes on a sidebar</p>
  </aside>
  <footer>Copyright 2010</footer>
</body>
</html>
```

Obrázek 2.4. Srovnání HTML4 s HTML5 zdroj: (Agarwal, 2011)

2.6. Nejpoužívanější Pluginy HTML5

Zde byly vybrány některé z nejpoužívanějších pluginů HTML5 za rok 2013.

2.6.1. Kontaktní formulář AJAX

Jedním z nejpoužívanějších HTML5 pluginů je kontaktní formulář AJAX. Velmi rychle a jednoduše umožňuje kontakt pomocí webu. Tento plně validní formulář využívá nejnovějších možností HTML5 kódu a CSS3 animace. Formulář nejen perfektně funguje, ale také skvěle vypadá. Obsahuje i funkcionality jQuery a anti-spam filtr. (Wolford, 2013)

2.6.2. Prezentační plugin CCSlider

Můžeme ho využít v portfoliích umělců, fotografů a kohokoliv kdo chce elegantní a plně interaktivní prezentační galerii. K dispozici je 14 3D přechodů a 16 2D přechodů, vše je opět ovládáno pomocí jQuery a HTML5. K dispozici je také nouzové přepnutí na 2D přechody kvůli starším prohlížečům, které nejsou schopny podporovat technologii Canvas. Je možno nastavit počet řezů používaných pro 3D efekty, přidávat popisky s animovanými přechody a mnohem více. (Wolford, 2013)

2.6.3. Hudební přehrávač MusicBox

MusicBox umožňuje vytvářet na stránkách seznam hudebních skladeb přístupný pomocí velmi sofistikovaného uživatelského rozhraní. Využívá to nejnovější co CSS3 nabízí k vytvoření aplikace bez obrázků, která je schopna přehrávat hudbu. Je také vybaven nouzovým přechodem na FlashPlayer, aby zajistil funkčnost ve starších prohlížečích. MusicBox získává informace o skladbách z každého hudebního souboru od názvů skladeb až po obal alba a tím přináší dokonalý jak poslechový tak obrazový zážitek. Jedinou slabou stránkou MusicBox je, že umí pracovat pouze s MP3 a OGG soubory. (Wolford, 2013)

2.6.4. Easy Video Background

Plugin Easy Video Background slouží k přidání videa na pozadí libovolné webové stránky pomocí jednoduchého JavaScript modulu a je v něm zahrnuta i podpora pro mobilní zařízení. Pomocí jediného volání JavaScriptu, tento plugin automaticky zvolí formát videa a načte video bez jakýchkoli potíží. Sám také automaticky rozpozná, když je stránka prohlížena z mobilního zařízení a zvolí nižší kvalitu videa pro snadnější přehrávání. Divák může ovládat videa na pozadí pomocí jednoduchého ovládání. (Wolford, 2013)

2.6.5. HTML5 Video & YouTube Background

Už z názvu je nejspíš patrné, o co v tomto pluginu kráčí. Tento plugin přehrává na pozadí nejen vaše lokální videa, ale také videa ze serveru YouTube. Tento video přehrávač je také optimalizován pro mobilní zařízení a tím poskytuje stejně kvalitní přehrávání pro stolní počítače i mobilní zařízení. Nejen že má podporu pro ID videí z YouTube a seznamu videí, ale nabízí také automaticky generované náhledy videí. (Wolford, 2013)

2.7.CSS

Cascading Style Sheets neboli CSS web designový jazyk, který se používá k naformátování webové stránky a celého jejího obsahu podle stylu prezentace zadaného autorem webové stránky. Tento způsob úpravy přináší obrovskou škálu výhod v oblasti formátování a prezentace prvků na webové stránce. CSS ve své podstatě odděluje obsah dokumentu od způsobu, ve kterém je daný dokument prezentován, což umožňuje mnohem plynulejší přechody a vysokou kompatibilitu mezi různými prohlížeči a platformami. CSS také umožňuje mnohem preciznější ovládání pro nastavení mezer, zarovnání a umístění obsahu, aniž by se muselo spoléhat na layoutové tabulky nebo rámy. Stylování písma, jeho barva i velikost písma mohou být taktéž velmi snadno ovládány pomocí CSS. (Gasston, 2011)

2.8.Historie CSS

Éra kaskádových stylů začíná v roce 1994 v CERNu, v místě zrození webu, v době, kdy web začíná být běžně používán jako platforma pro elektronické publikování. Ovšem zanedlouho vyvstane problém a to, že zásadní součást publikační platformy chybí. Zatím neexistuje žádný způsob, jak stylovat dokumenty. Například nebyl doposud vyvinut žádný způsob, jak popsat na webové stránce layout novin a jiné podobné layouty. Pionýr v oblasti webu Håkon Wium Lie, který dříve pracoval na personalizovaných novinových prezentacích v MIT Media Laboratory, si poprvé uvědomil nutnost stylových šablon pro web. (Håkon Wium Lie, 1997)

Používání stylů v prohlížečích nebyla úplně nová myšlenka. Oddělení struktury dokumentu od layoutu dokumentu bylo cílem HTML už od počátků v roce 1990. Tim Berners Lee napsal svůj hybrid editoru a prohlížeče jménem NeXT takovým způsobem, aby mohl nastavit styl pomocí velmi jednoduchého stylování. Ovšem nezveřejnil syntaxi stylování protože si myslel, že prohlížeč sám by měl rozhodovat o nejvhodnějším a nejpřijatelnějším zobrazení stránky uživatelům. Ostatní prohlížeče většinou obsahovaly podobné stylové jazyky. (Håkon Wium Lie, 1997)

Místo toho, aby vznikalo více pokročilejších stylových šablon, nabízely následující prohlížeče svým uživatelům naopak stále méně a méně možností, jakým způsobem mohli styly ovlivnit. Pisatelé webových stránek si začali stěžovat, že nemají dostatečně velký vliv na to, jak jejich stránky budou vypadat. Jednou hlavních otázek, od autorů nově vznikajících webů bylo, jak změnit písma a barvy prvků. HTML však v té době ještě neobsahoval kýženou funkcionalitu. (Håkon Wium Lie, 1997)

Tři dny předtím než firma Netscape oznámila dostupnost svého nového prohlížeče, Håkon oficiálně publikoval první verzi kaskádových stylů v HTML. Mezi tím Dave Raggett (hlavní architekt HTML 3.0) podpořil vydání návrhu CSS ještě před nadcházející konferencí v Chicagu. Dave došel k názoru, že HTML by se nikdy nemělo proměnit v jazyk stylizace a popisů stránky, a že je nutné vytvořit víceúčelový mechanismus pro splnění požadavků vyplývajících z přání autorů. Ačkoli první verze dokumentu ještě zdaleka nebyla ideální, byla základním krokem pro budoucí rozšíření a využívání kaskádových stylů. (Håkon Wium Lie, 1997)

2.9.CSS3

CSS3 je nejnovější verze stylovacího jazyka CSS. V CSS3 nejde jen o odkázání na nové funkce CSS, ale jedná se o třetí úroveň pokroku ve specifikaci standardu CSS. CSS3 zahrnuje veškeré vymoženosti a funkce předchozích verzí CSS, ale také spoustu nových funkcí a prvků, které se dají využít k co nejkvalitnější implementaci stylů, aniž bychom museli použít složité skripty, nesémantické stylování nebo zbytečné obrázky. (Hogan, 2011)

Mezi nově vytvořené funkce CSS3 patří například podpora nově přidáných selektorů, stíny a stínování, zaoblené rohy, více pozadí, animace, průhlednost a spoustu dalších užitečných funkcí. (Hogan, 2011)

2.9.1. Selektory

Obecně se selektory dají rozdělit na 2 skupiny. Selektory, které ovlivňují přímo prvky, které jsou definované v stavbě stromu dokumentu. Například p prvky a href atributy atd., tato skupina obsahuje třídové, typové a atributové selektory. V druhé skupině nalezneme pseudo-selektory, které ovlivňují prvky a také informace, které nejsou formální součástí stromu dokumentu. Prvky jako první písmeno odstavce nebo posledního následníka rodičovského prvku. CSS3 nabízí tři nové typy selektorů atributů, které můžeme dle libosti využít a jeden nový kombinátor. Kombinátor je v podstatě selektor, který umí spojit jiné selektory dohromady. Tyto selektory jsou podrobně definovány v modulu selektorů třetí úrovně, který

má nyní status navrhovaných doporučení. Tento status znamená, že modul je již světově rozšířen a stabilně implementován ve velkém množství běžně používaných internetových prohlížečů. (Gasston, 2011)

Atributové selektory v CSS3

Počátek atributových selektorů byl v CSS2, jak už jméno jasně napovídá, umožňují uživateli určit pravidla, která odpovídají prvkům na základě jejich atributů. Při tvorbě CSS3 byla tato vlastnost ještě dále rozvinuta. Selektory v novém CSS3 dokáží dokonce, aby odpovídaly podřetězce v rámci hodnot atributů. Tato nová vlastnost je dělá obzvláště užitečnými pro použití pravidel v XML dokumentech, jelikož XML dokumenty mají mnohem více libovolných atributů než HTML. Ovšem i přes tento fakt jsou pro vývojáře HTML stále velmi užitečné. (Gasston, 2011)

Pseudo-třídy a pseudo-elementy

Už první specifikace CSS nám představila pojetí pseudo-tříd a pseudo-prvků. Jsou to selektory, které působí na informace o prvcích sedících mimo strom dokumentu nebo přesahující strom dokumentu, jako je například první písmeno textového oddílu. Pseudo-třída nám umožňuje rozlišit různé stavy nebo druhy prvku. Pseudo-element nám poskytuje přístup k dílčí částem prvku, zde patří ty pseudo-prvky, které zahrnují části textových oblastí. Přestože tyto selektory existovaly už od prvních verzí CSS, CSS3 přidala velké množství nových selektorů. Ačkoli stále platí, že podpora pseudo-prvků ještě pořád není úplně realizována a implementována. Prohlížeče podporující tuto možnost se začaly objevovat až v poslední době. CSS3 je postavena na těchto základech ovšem s podstatně rozšířeným rozsahem pseudo-tříd a také s mírně vylepšenou syntaxí pro dobré rozlišení pseudo-prvků. (Gasston, 2011)

2.9.2. Barvy

Před vytvořením CSS3 bylo stylování barev možné nastavit pouze pomocí hexadecimálních kódů barev jako je například 000000 pro černou barvu nebo FFFFFFFF pro bílou. Další možností pro zadávání barev bylo pomocí jejich RGB notace buď pomocí celých čísel, nebo pomocí procentuálního poměru. Některé ze základních barev bylo možno reprezentovat klíčovými slovy jako např. red, blue, black. CSS3 tuto paletu podstatně rozšířila o 147 barev, které jsou povětšinou velmi dobře podporovány prohlížeči. Je zde také úplně nová možnost využití i jiných barevných modelů, které dosud nebyly podporovány jako

HSLA, HSL a RGBA. Ovšem nejznatelnějším krokem kupředu je zavedení transparentních barev, což přineslo do CSS obrovskou škálu nových možností.

RGBA

RGBA je ve své podstatě klasický RGB model, který je ale rozšířen o novou hodnotu jménem Alfa kanál. Alfa kanál vyjadřuje průhlednost. Z toho plyne, že můžeme využít klasického modelu červená, zelená, modrá a k tomu ještě hodnotu alfa, která může nabývat 0, 1 a jakéhokoli čísla z intervalu 0 – 1. Hodnota 0 znamená úplně transparentní a hodnota 1 naopak znamená úplně neprůhledný. Na rozdíl od RGB neexistuje pro RGBA hexadecimální zápis. Tato možnost byla nějakou dobu zvažována, ale nakonec se od ní upustilo. Používá se tedy zápis typu 0,0,0,0.8, což ve výsledku znamená čistě černá s 20% průhledností. Nicméně je neustále důležité zadávat barvy obojím způsobem jak RGB tak RGBA notací, jelikož starší prohlížeče RGBA neumí používat. Hlavně starší verze Internet Exploreru. (Goldstein, 2011)

HSLA

Vychází z klasického HSL modelu a stejně jako u RGB a RGBA, je HSLA oproti HSL rozšířen o Alfa kanál, takže se opět jedná o průhlednost. 0 znamená úplně transparentní a 1 úplně neprůhledný. HSL model vychází ze způsobu, jakým vnímá barvu lidské oko, tj. pomocí 3 faktorů - odstínu (hue), sytosti (saturation) a světlosti (lightness). (Goldstein, 2011)

Odstín se zobrazuje na kruhu od 0 do 359 stupňů. Sytost se vyobrazuje jako procentuální hodnota. Standardní 100% sytost znamená, že bude mít barva svůj plný odstín, 0% sytosti znamená, že bude mít barva odstín šedi. Co se světlosti týče tak je také procentuálně vyjádřena a její standardní hodnota je 50%. Při hodnotě 100% dostaneme úplně bílou a při hodnotě 0% dostaneme úplně černou. (Goldstein, 2011)

Pro hodně lidí je tento způsob tvorby barev intuitivní, dobře se jim s ním pracuje a umožňuje velmi rychlé drobné úpravy barev. Každopádně musíme počítat s tím, že pokud chceme, aby naše stránky dobře fungovaly na starších verzích Internet Exploreru, je třeba vždy používat klasický hexadecimální zápis.

2.9.3. Zaoblení rohů

Další novinkou v CSS3 možnost jak dělat zaoblené rohy bez pomoci grafických editorů. Takže to ušetří spoustu času při tvorbě a i při budoucích úpravách. Zaoblení se nastavuje pomocí nového parametru border-radius například:

```
border-radius: 10px;
```

(Goldstein, 2011)

Toto je ovšem zjednodušená verze zápisu pokud chceme všechny 4 rohy stejné. Je samozřejmě možné pomocí použití dodatečných parametrů top-left, top-right, bottom-left, bottom-right, nastavit různé zaoblení pro každý roh. Ve starších verzích Firefoxu se používá prefix moz a parametry topleft, topright, bottomleft, bottomright. (Goldstein, 2011)

Takto by tedy vypadala syntaxe pro starší verze Firefoxu.

```
-moz-border-radius-topleft: 8px;  
-moz-border-radius-topright: 10px;  
-moz-border-radius-bottomright: 20px;  
-moz-border-radius-bottomleft: 40px;
```

(Goldstein, 2011)

A takto klasický zápis pro novější verze.

```
border-top-left-radius: 5px;  
border-top-right-radius: 8px;  
border-bottom-right-radius: 12px;  
border-bottom-left-radius: 30px;
```

(Goldstein, 2011)

2.9.4. Stínování

CSS3 umožňuje přidání stínů pomocí vlastnosti box-shadow. Box-shadow nám umožňuje zadat barvu, výšku, šířku, rozostření a posun jednoho nebo více vnitřních či vnějších stínů prvků. Obvykle je na stíny pohlíženo jako na efekt, který budí v prvcích dojem prostorovosti. Dojem jakéhosi vznášení se nad povrchem stránky. Nicméně díky široké škále možností, které nyní CSS3 nabízí je možno experimentovat mnohem více. Můžeme například použít možnost stínu s nulovým rozostřením. Místo stínu nám tak vznikne spíše 3D prostorový efekt. Box-shadow může obsahovat čárkami oddělený seznam stínů, který funguje jako jeho hodnota. Stíny mají obvykle dvě až čtyři hodnoty velikosti, barvu a klíčovou hodnotu inset. Pokud inset není nastaven, výchozí stín se objeví směrem ven z objektu. Nyní se podíváme, co který parametr stínu dělá. (Goldstein, 2011)

```
-webkit-box-shadow: 3px 5px 0 0 rgba(72,72,72,1);  
-moz-box-shadow: 3px 5px 0 0 rgba(72,72,72,1);  
-box-shadow: 3px 5px 0 0 rgba(72,72,72,1);
```

(Goldstein, 2011)

První hodnota v pořadí značí horizontální posun. Pokud je kladná, stín se vytvoří napravo od prvku a pokud je záporná, tak nalevo od prvku. (Goldstein, 2011)

Druhá hodnota je vertikální posun. Pokud je kladná, pohybuje se stín dolů pod prvek a pokud je záporná, stín se naopak pohybuje nahoru směrem nad prvek. (Goldstein, 2011)

Pokud zadáme i třetí hodnotu, jedná se o rozsah rozostření stínu. Jsou zde povoleny jen pozitivní hodnoty, a čím větší jsou, tím více je stín rozostřen. (Goldstein, 2011)

Čtvrtá hodnota nastavuje vzdálenost stínu. Pokud je kladná, tvar stínu se začne rozšiřovat, pokud záporná, tvar stínu se smršťuje. (Goldstein, 2011)

Poslední hodnotou je barva, která jednoduše nastavuje barvu stínu. Pokud tuto možnost vynecháme, výchozí barva by měla být stejná jako vlastnost color příslušného elementu. Například Firefox tento fakt bez problému podporuje, ovšem třeba WebKit ne, proto je třeba vždy barvu explicitně zadávat, abychom se vyhnuli nepříjemnostem. Je možné klidně použít i hexadecimální zápis barvy, ale ve většině případů použijeme notaci HSLA nebo RGB, jelikož stíny bývají částečně průhledné. (Goldstein, 2011)

2.10. Prvek Canvas

Canvas neboli plátno přináší hodně nového a konečně osvobozuje HTML od kreslení jednoduchých tvarů. Umožňuje totiž kreslit v podstatě cokoli za pomoci JavaScriptu. Tato vlastnost velmi usnadňuje načítání stránek z webu, jelikož už počítač nemusí stahovat obrovské množství obrázků, jako tomu bylo doposud. Pomocí plátna se dá vyobrazit nespočet tvarů, křivek, výplní, přechodů, atd. a také může sloužit k vyobrazení videa. (Goldstein, 2011)

2.10.1. Historie Canvas

Plátno bylo vytvořeno firmou Apple a je postaveno na základech frameworku Quartz 2D, což byl už delší dobu framework používaný Apple na vykreslování 2D obrazů. S výtvořem plátna byly některé možnosti z Quartz 2D posunuty ještě dál. Poté začalo být využíváno prohlížeči Opera a Mozilla a nakonec bylo přijato W3C při tvorbě standardů HTML5. (Goldstein, 2011)

2.10.2. Vytvoření Canvas

Nejprve je třeba vytvořit na naší stránce prvek canvas.

```
<canvas>
Nejsem podporován.
</canvas>
```

Pokud váš prohlížeč plátno nepodporuje, uvidíte vyobrazen text uvnitř tagů. (Goldstein, 2011)

Jelikož plátno musí být ovládáno JavaScriptem je třeba mu dát jeho vlastní id.

```
<canvas id="Platno">
Nejsem podporován.
</canvas>
```

Poté je také dobré zadat našemu plátnu atributy width="xxx" a height="xxx", které jasně definují jeho výšku a šířku. V neposlední řadě je vhodné udělat v našem CSS souboru nastýlování plátna, tzn. přidat alespoň nějaké ohraničení, jelikož plátno jako takové není na stránce viditelné a atribut border nám plátno zviditelní a zpřehlední. Nakonec je třeba propojit plátno s JavaScriptovou funkcí jQuery, která se bude starat o samotné nakreslení. (Goldstein, 2011)

2.10.3. Kreslení pomocí Canvas

Jak už bylo řečeno samotné kreslení na plátno je realizováno pomocí API JavaScriptu. Musíme tedy použít funkci s názvem draw(), kterou musíme zadat do našeho script prvku, tzn. mezi tagy <script> </script>. Nejdříve si musíme uložit plátno do proměnné. (Goldstein, 2011)

```
<script>
:
function draw() {
var canvas = document.getElementById("Platno");
}
</script>
```

(Goldstein, 2011)

Následně si musíme nastavit kontext, ve kterém bude naše plátno kresleno. Dnes je většinou všeobecně podporován jen 2D rozměr. CanvasRenderingContext2D objekt specifikuje tento

kontext dle standardu W3C. Kontext získáme pomocí metody `getContext` a přiřadíme jí stringovou hodnotu „2d“. Nyní můžeme kreslit ve dvou dimenzích. (Goldstein, 2011)

```
function draw() {  
var canvas = document.getElementById("Platno");  
var context = canvas.getContext("2d");  
}
```

(Goldstein, 2011)

Poté musíme „namočit náš štětec do barvy“ stejně jako při kreslení na reálné plátno, což se ve virtuálním plátně dělá pomocí možností `strokeStyle` a `fillStyle`. Obě tyto možnosti se nastaví na kontext objektu. Obě mají 3 hodnoty:

- String, který reprezentuje barvu,
- `CanvasGradient`,
- `CanvasPattern`.

3. Analýza současného stavu webů konkurenčních výrobců kytar

V této kapitole se budeme zabývat weby, které používají podobný typ aplikací jako je námi vytvářená. Bude brán zřetel na kladné a záporné stránky konkurenčních firem. Jinými slovy se budeme snažit poukázat na to, co mají konkurenční webové aplikace lepší, co bychom si tedy od nich měli vzít za příklad a naopak co se konkurenci příliš nepovedlo a co bychom mohli udělat lépe. Budeme se zde také věnovat základnímu seznámení s firmou, která nás oslovila, jejím požadavkům, představám a firemní situaci.

Webových aplikací podobného ražení sice moc není, ale některé firmy z odvětví výroby hudebních nástrojů již podobné aplikace úspěšně používají. Problém je ovšem v tom, že tak jako jsou produktové řady jednotlivých firem velmi různorodé a navzájem odlišné, tak se tato odlišnost projevuje i v jejich aplikacích a možnostech, které nabízejí. Neméně se liší i technologie používané k výrobě těchto aplikací. Metoda, která byla zvolena pro naši aplikaci je použití klasického HTML v kombinaci s JavaScriptem a CSS stylováním. Některé konkurenční firmy ovšem pro své aplikace zdárně používají například Flash Player. A můžeme nalézt i úplně jinak pojaté koncepce.

3.1. Firma Warmoth

The screenshot shows the 'Begin building your Warmoth custom built' interface. At the top, there's a 'Start HERE' button and a 'Stratocaster REPLACEMENT' label. The main section is divided into three columns for different construction types: 'CONSTRUCTION Warmoth PRO', 'VINTAGE MODERN', and 'TOTAL VINTAGE'. Each column lists features and has a 'DETAILS' button. To the right, there are 'SCALE CONVERSIONS' for 'PRO 24 3/4\"'

Obrázek 3.1. Aplikace firmy Warmoth zdroj: (www.warmoth.com)

Americká firma Warmoth, jejíž stránky můžeme najít na adrese www.warmoth.com, je jednou z nejznámějších světových firem zabývajících se CNC výrobou kytar a kytarových dílů, které začaly podobou aplikaci používat. Jedná se však o velkovýrobu, takže se nedá říct, že by to byl přímý konkurent malé české firmy. Nicméně se z jejich aplikace můžeme hodně poučit při tvorbě naší vlastní aplikace.

Aplikace firmy Warmoth svými možnostmi několikanásobně překračuje rámec rozsahu potřeb pro námi tvořenou aplikaci. Rozsáhlé možnosti jsou určitě velké plus, ale problémem je orientace. Možnosti jsou tak rozsáhlé, že většina zákazníků musí mít často problém najít a vybrat to, co zrovna hledá a potřebuje. Nehledě na to, že údržba tak

složitého a rozsáhlého JavaScriptového kódu není zrovna jednoduchou záležitostí a někdy i sebemenší změna zabere několik hodin času.

Vzhled uživatelského rozhraní je sice v celku přijatelný, stejně jako layout celé stránky, ale problém nastává v situaci, kdy by chtěl potenciální zákazník vidět svůj vytvořený nástroj v celku. Firma se totiž evidentně specializuje na klientelu, která nechce skládat celé nástroje, nýbrž jen jejich součásti. Toto rozhodně není pro naše potřeby vhodné, jelikož firma, která nás oslovila, doposud vyráběla nástroje hlavně ručně a s postupným přechodem na CNC technologie chce v budoucnu začít vyrábět i standardizovanější modely, u kterých ovšem budou mít zákazníci možnost úpravy.

Nicméně formát velkého množství výběrových roletek s možnostmi je určitě dobrým řešením a měli bychom se jím inspirovat i v naší aplikaci. K nim rozhodně přidáme i možnost zobrazení jednotlivých prvků, jelikož jen volit nestačí. Pro zákazníka je důležité i v reálném čase vidět výsledek, aby měl ucelenou představu.

Co se týče možnosti vyobrazení ceny jednotlivých úprav a následně výsledné ceny produktu, tato možnost je určitě dalším kladem firmy Warmoth a taky se jí inspirujeme.

Layout, jak už bylo zmíněno výše, je celkem přehledný a intuitivní. To, co dělá aplikaci nepřehlednou, není ani tak layout, ale spíše přehnané množství možností. Někomu by mohla vadit tmavost a šedočerné kontury celého layoutu, ale i přes tyto detaily plní layout svůj účel velmi dobře.

3.2. Firma Usa Custom Guitars



Obrázek 3.2. Aplikace firmy Usa Custom Guitars zdroj: (www.usacustomguitars.com)

Další aplikace, která stojí za zmínku, je aplikace od firmy Usa Custom Guitars, jejíž webové stránky můžeme najít na adrese www.usacustomguitars.com. Tato firma pojala celou aplikaci na stavbu kytar dost odlišně. Mají vytvořeno několik verzí a každá se týká jiného kytarového modelu.

Celá aplikace je dělaná ve Flash Playeru, což sice přináší spoustu výhod, ale také velkou škálu nevýhod.

Aplikace má obrovské množství možností, ale problémem je, že kvůli tomuto obrovskému množství prvků aplikace neuvěřitelně dlouho načítá. Toto může být u potenciálních klientů s menší rychlostí internetového připojení celkem problematické a spousta z nich to může odradit hned na samém začátku. Dá se předpokládat, že kvůli tomuto problému raději firma sama rozdělila aplikaci na 3 různé aplikace pro každý model zvlášť. Při spojení do jedné velké aplikace by byla doba načítání ještě delší.

Dalším negativem aplikace této firmy je poněkud nepřehledné menu, které i na druhý pohled není často úplně srozumitelné a jasné a dá se v něm snadno ztratit. Nehledě na to, že i samotné přepínání položek v menu je celkem zdlouhavou záležitostí, což je po čekání na načtení aplikace další nepříjemnost.

Za vyzdvižení stojí obrovská škála možností, které jsou téměř nevyčerpatelné. Mimo jiné například neuvěřitelně rozsáhlá paleta barev a možnost používání barevných přechodů tzv. burstů.

Dalším zajímavým prvkem je možnost použití URL vlastního obrázku jako grafiky těla kytary. Sice je velmi nepravděpodobné, že by byla firma schopna tuto grafiku na tělo vyrobit, ale rozhodně je to příjemným zpestřením oproti fádním obyčejným barvám.

Nakonec bychom měli zmínit jeden velký nedostatek. Aplikace vůbec není prodejně orientovaná. Takže nikde v celé aplikaci není možno najít jakoukoli zmínku o ceně popř. o možnosti produkt koupit. Jedná se tedy vlastně jen o aplikaci pro nadšence, které baví si svou kytaru pro představu složit, což ve výsledku může vést ke koupi dané kytary, ale taky nemusí.

3.3. Firma Fender



Obrázek 3.3. Aplikace firmy Fender zdroj: (www.fender.com)

Tato firma je pro mnohé synonymem pro elektrickou kytaru. Jedná se o jednu ze světově nejznámějších a nejrozsáhlejších značek, která už od roku 1946 vyrábí světoznámé elektrifikované nástroje. Proto se zde určitě zmíníme i o jejich aplikaci na online návrh kytary. Web firmy najdeme na adrese www.fender.com.

Aplikace, kterou firma Fender vypustila pod názvem Configurator, by se dala shrnout i jednou větou: „V jednoduchosti je síla“. Ač se jedná o světoznámou renomovanou firmu, jejich Configurator je velmi jednoduše řešený.

Layout i menu celé aplikace je naprosto intuitivní. Podobně jako Usa Custom Guitars si na začátku zákazník musí vybrat, pro který ze čtyř nejznámějších nástrojů této legendární firmy chce aplikaci spustit.

Možnosti sice nejsou zdaleka tak široké jako u výše zmíněných aplikací, ale rozhodně to není na škodu. Je evidentní, že aplikace je dělaná pro jednoduché pochopení, pro jasný a přehledný výběr standardizovaných prvků. Jinými slovy: „Co si vyberete, to vám bez problému postavíme a odešleme až domů.“

Fender samozřejmě nijak neopomíná možnost okamžité koupě nástroje hned po dokončení výroby a u každého zvoleného prvku je jasně napsána jeho cena, která se během „stavby“ kumuluje v ukazateli price. Tento způsob se zdá být jako velmi vyvedený a určitě si z něj vezmeme ponaučení i pro naši aplikaci.

Třešničkou na dortu aplikace firmy Fender je možnost zoomu vyobrazení nástroje, takže je možné se podívat na poskládanou kytaru i velmi zblízka. Není to sice nijak zvlášť důležitý prvek, ale rozhodně vypadá zajímavě. Dále je zde také velmi příjemná možnost vybrat si při poslední fázi výroby i typ a barevné provedení kufru vhodného pro nástroj.

Úplně nakonec je ještě velmi zajímavým prvkem možnost přímého sdílení vytvořeného nástroje pomocí sociálních sítí Facebook, Google+, Twitter a Pinterest.

3.4. Moniker Guitars



Obrázek 3.4. Aplikace firmy Moniker Guitars zdroj: (www.monikerguitars.com)

Poslední aplikací, kterou se budeme v této části zabývat je aplikace firmy Moniker Guitars, jejíž web najdeme na adrese www.monikerguitars.com.

Tato aplikace je opět velmi zdařilá a opět se drží hesla „V jednoduchosti je síla“. Oproti aplikaci od firmy Fender má pár zajímavých možností, které stojí za zmínku.

Provedení je velmi podobné jako u výše zmíněné firmy Fender, ale přibyl zde nový prvek vlastní grafiky, který je velmi zajímavý. Moniker Guitars vám nabízí paletu motivů ve vektorovém formátu (různé typy obrazců, plamenů, hvězd atd.), které je možno si libovolně zvětšit, otočit, změnit jim barvu a umístit si je kdekoli na kytaru, což přidává obrovské množství nových vzhledových možností.

Zbytek klasických možností jako výběr hardware, typu dřeva atd. je už poněkud slabší, ale stále postačující.

Jednou z negativních vlastností je, že většina prvků, které jsou nezbytné, je už započítána v původní ceně. Zákazník nemá úplný přehled o tom, co kolik stojí, s výjimkou komponent, u kterých se cena kus od kusu může výrazně lišit (jako například snímače).

3.5. Analýza stavu firmy

Firma, pro kterou bude aplikace tvořena, je malá česká firmička o deseti zaměstnancích. Doposud vyráběla firma elektrické kytary vyloženě za pomoci ruční výroby a ručních elektrických přístrojů. Od konce roku 2012 však firma začala postupně přecházet na výrobu některých prvků pomocí CNC strojů a nyní chce kromě vysoce specifických zakázkových modelů začít i s výrobou několika standardizovanějších modelů. Bude se jednat o modely, které sice budou standardizované, ale přesto na nich bude dost velká možnost customizace ze strany zákazníků.

Z toho vznikl nápad, po vzoru zahraničních kolegů, na vytvoření aplikace pro grafický návrh a úpravy daného nástroje zákazníkem samotným. Firma má vlastní webové stránky a proto si logicky přála, aby bylo možné přistupovat k aplikaci pomocí těchto webových stránek. Web firmy sice není úplně zdařilý, ale firma sama si ho zatím nepřeje měnit, pouze do něj zařadit onu webovou aplikaci.

Firma, ve snaze jít s dobou, také začala používat profil na sociální síti Facebook, ale podle vlastních slov vedení firmy zatím nelze pozorovat žádný extra velký zájem ze strany uživatelů Facebooku. Nicméně firma je na Facebooku teprve krátce, takže to vše může být jen otázkou času.

Hlavní požadavky ze strany firmy tedy byly

- Jednoduchost,
- Přehlednost,
- Pěkný vzhled,
- Grafické rozhraní,
- Obsažení všech firmou nabízených prvků,
- Přepínání jazyka čeština/angličtina,
- Možnost okamžité objednávky,
- Odeslání objednávky na mail firmy,
- Obsažení všech zvolených parametrů nástroje v objednávce.

Snažili jsme se těmito požadavkům co nejlépe vyhovět a zajistit jak funkčnost, tak i spokojenost firmy.

4. Návrh internetového rozhraní

Po pečlivém prostudování podobných internetových rozhraní konkurenčních výrobců kytar jsme vybrali všechny pozitivní aspekty konkurenčních aplikací a pokusíme se je implementovat i v našem rozhraní. Jelikož konkrétní technologie, kterou máme použít, firmou stanovena nebyla. Bylo pouze řečeno, že se k aplikaci musí přistupovat z internetu a bylo by vhodné, aby byla v dvojjazyčném provedení s možností přepínání mezi českou a anglickou verzí. Rozhodli jsme se tedy pro kombinaci technologií JavaScriptu a HTML, takže naše rozhraní či aplikace bude ve své podstatě webová stránka s velkým množstvím scriptových příkazů. Jednou z dílčích technologií bude Canvas, což je nový prvek HTML5 a budeme jej využívat na grafické zobrazení výsledného produktu na plátně v reálném čase. Aplikace bude vytvořena tak, aby v budoucnu mohla být s minimálními úpravami použita i na webových stránkách jiných firem z tohoto odvětví, které by o ni projevíly zájem.

4.1. Layout stránky

Budeme se snažit vycházet ze zvyklostí a z toho, co od nás potenciální návštěvníci budou očekávat. Firma sama předpokládá, že většina zákazníků budou hudebníci hrající rockovou popř. metalovou muziku, takže všeobecně spíše tvrdší hudební žánry. Přihlédneme tedy k tomuto faktu i při návrhu layoutu a budeme se snažit udělat ho v temnějších kovových barvách v kombinaci s texturami dřevěných motivů, která by muzikanty tohoto ražení mohla zaujmout.

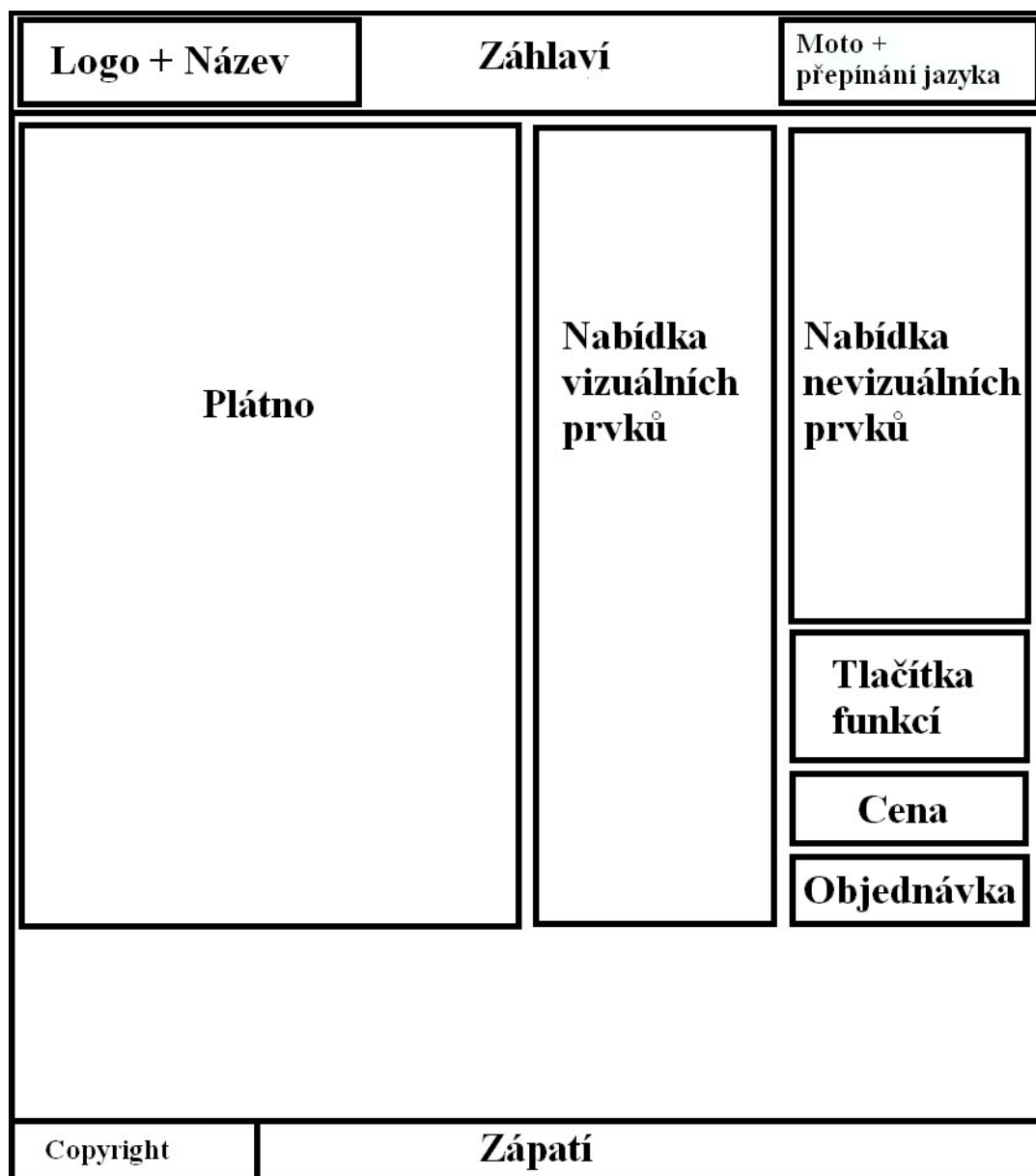
Budeme hojně využívat grafických prvků, které nabízí HTML5 jako jsou gradienty, stíny textu, stíny prvků, zaoblení rohů atd. To vše proto, abychom redukovali nutnost použití textur v samotném layoutu na minimum. Obrovské množství textur bude totiž třeba pro jednotlivé prvky nástroje.

Každý prvek vyžaduje skupinu obrázků vyřezaných z pozadí a ještě k tomu často v několika různých barvách. Pro tyto účely použijeme formát PNG.

Velikost stránky bude optimalizována pro dnes už relativně běžné rozlišení širokoúhlých monitorů 1920 x 1080 pixelů. Majitelé menších a starších monitorů budou tedy nutně muset používat posuvník a naopak majitelé moderních monitorů s extrémně velkým rozlišením budou nejspíš muset použít přibližovací schopnosti jejich webového prohlížeče.

Nicméně v globálním měřítku je námi zvolená zlatá střední cesta asi tou nejschůdnější variantou.

Dalším aspektem je, že by se stránka měla koncepčně lišit od zbytku webu, aby bylo zákazníkovi na první pohled jasné, že se dostal do nové funkční možnosti stránek firmy. Z tohoto důvodu se bude stránka také ve výchozím nastavení otvírat do nového panelu prohlížeče.



Obrázek 4.1. Wireframe layoutu zdroj: (vlastní obrázek)

4.1.1. Záhloví a zápatí

Prvek záhlaví bude využit velmi jednoduše čistě k vyobrazení názvu aplikace aplikačního loga. Bude v něm použita také textura pro zkrášlení aplikace. Prvek zápatí bude sloužit pouze pro vyobrazení copyrightu.

4.1.2. Canvas

Na stránce tedy bude umístěno plátno Canvas, pro jeho umístění jsme zvolili levou stranu stránky po vzoru několika konkurenčních firem. Důvodem je nejen konkurence, ale i logičnost celé aplikace. Zákazník bude intuitivně soustředit svůj pohled do levé části stránky, a proto by bylo vhodné, aby se grafické prvky kytary zobrazovaly nalevo.

4.1.3. Navigace

Pro tvorbu navigace využijeme prvku <nav>, který nabízí HTML5. Navigace bude tedy vlastně přehled nastavitelných prvků pro tvoření kytary. Bude obsahovat výběrové roletky, ve kterých si budeme moct jednotlivé prvky vybrat a JavaScript následně provede jednotlivé činnosti.

Co se pozice navigace týká, bude netradičně vpravo. Respektive netradičně pro většinu webových stránek, ale v celku tradičně pro podobný typ aplikací jako je ta naše. Je kladen důraz na to, aby se v levé sekci stránky zobrazovaly grafické prvky skládaného nástroje. Proto bude menu s výběrem orientováno na pravou část stránky.

Po delší úvaze jsme se rozhodli rozdělit výběrové prvky do dvou skupin.

- Vizualní prvky, které se projeví na grafickém aspektu nástroje, tzn. prvky, které lze na plátně po vykreslení vidět. Např. barva těla kytary nebo typ kobylky na kytáře.
- Nevizualní prvky, které jsou sice důležitou součástí výběru, ale v 2D kontextu nejsou viditelné a nelze je tak na plátně vykreslit. Např. typ dřeva použitého na výrobu krku, rádius zaoblení hmatníku, výška pražců atd.

Pro každou z těchto skupin bude logicky vytvořena vlastní navigace, takže budou na stránce ve své podstatě sousedit dvě navigace v pravé části přímo vedle sebe.

4.1.4. Tlačítka

Bude taky využito několik druhů tlačítek, které budou provádět různé funkce dle potřeby. Jako například tlačítko na odeslání celého výběru tzn. vytvoření objednávky. Nebo tlačítko na vyčištění všech doposud nastavených možností atd.

4.1.5. Cena

Je také nutné vyhradit prostor pro část stránky, ve které se bude vyobrazovat kumulovaná cena zákazníkem zvolených prvků. Pro tento účel jsme zvolili pravý dolní roh stránky přímo pod nabídkami potažmo tlačítka.

5. Řešení a implementace rozhraní

Řešení a realizace celého rozhraní proběhla tedy pomocí klasické HTML5 stránky a JavaScriptového kódu, který běží celou dobu na pozadí. Kód není přímo součástí stránky, ale je obsažen v samostatné knihovně, kterou jsme nazvali main.js. Tuto knihovnu naše stránka importuje jako zdrojový soubor a využívá tak jejich funkcí. K podrobnostem týkajícím se praktického použití CSS stylů, prvků HTML a JavaScript kódu pro určité operace se budeme věnovat v této části.

5.1. Realizace layoutu

Co se layoutu týče, jak už bylo zmíněno v návrhu, v těle stránky používáme prvky `<header>`, `<footer>`, třináct navzájem se překrývajících prvků `<canvas>`, z nichž jen jeden je viditelný a ostatních dvanáct je plně transparentních. Dále navigační prvky `<nav>` a také několik tlačítek a nespočet divů.

5.1.1. Body

Tělo je nastylováno velmi jednoduše.

```
body {
background: linear-gradient(black,gray);
font: 100% Arial, Helvetica, sans-serif;
text-align: center;
padding: 0;
margin: 0 auto;
width: 1000px;
height: 1080px;
border: solid 3px black;
border-radius: 5px;
}
```

Jako pozadí je používán lineární přechod z černé do šedé barvy a jelikož byl záměr optimalizovat celou aplikaci pro rozlišení 1920 x 1080 pixelů, tak jsme zvolili šířku 1000px a výšku 1080px, aby při 100% zobrazení aplikace překrývala na výšku celou obrazovku a po stranách byl prostor.

5.1.2. Header

Záhlaví je na naší stránce velmi jednoduché, slouží pouze k vyobrazení loga aplikace, jejího názvu a banneru, který je vytvořen z textury slzového plechu. V oblasti hlavičky se sice zobrazuje také moto aplikace a ikona k přepínání jazykových verzí, ale ty jsou už v samostatném divu nazvaném catchphrase.

```
header{
    background-image: url(img/header.PNG);
    color: white;
    text-align: left;
    height: 80px;
    font-size: 36px;
    font-weight: bold;
    text-shadow: 2px 2px black;
    font-family: Copperplate ghotic bold;
    border: solid 3px black;
}
#catchphrase{color: white;
    text-align: right;
    font-size: 16px;
    margin-left: 760px;
    position: absolute;
}
```

5.1.3. Footer

Zápatí stránky není nijak složité a slouží pouze jako prostor pro zveřejnění copyrightu, takže má velmi jednoduché formátování černého pozadí s bílým textem a zarovnání textu na levou stranu.

5.1.4. Canvas

U všech pláten je deklarována vlastnost z-index:, která umožňuje nastavit prioritu při vykreslování. Čím větší je index, tím výše je plátno v hierarchii překrývání. Takže přirozeně nejspodnější plátno má nejnižší z-index 1. Všem plátnům jako ostatně většině prvků na stránce byly přiřazeny hodnoty id pro jasnou identifikaci pláten. Hodnoty jsou myCanvas1 až 13 a jsou velmi důležité pro samotné vykreslování. Všechna plátna jsou v CSS nastýlována

jak zvlášť kvůli různému z-indexu a viditelnosti, tak ve společném stylu pro prvek canvas. Vzhledem k tomu, že se styly dodatečných pláten opakují, ukážeme si jen prvních pět.

```
#Canvas{
    position: absolute;
    margin-left: 30px;
}
#myCanvas1 {
    z-index: 1;
    position: absolute;
    border: solid 2px white;
    background-color: grey;
    border-radius: 5px;
    color: white;
    background: linear-gradient(gray,black);
    box-shadow: 5px 5px 5px black;
}
#myCanvas2 {
    z-index: 2;
    position: absolute;
    border-radius: 5px;
}
#myCanvas3 {
    z-index: 3;
    position: absolute;
    border-radius: 5px;
}
#myCanvas4 {
    z-index: 4;
    position: absolute;
    border-radius: 5px;
}
#myCanvas5 {
    z-index: 5;
    position: absolute;
    border-radius: 5px;
}
...
```

5.1.5. Nav

Dalšími prvky na stránce jsou prvky <nav>, které má naše stránka dva. Jeden pro roletky, které se vizuálně projeví na stavbě kytary a druhý pro prvky, které nejsou vizuálně

poznat, ale přesto jsou to důležité specifikace nástroje. Co se layoutu navigací týče, snažili jsme se je umístit do pravé části stránky, aby byl nástroj vyobrazován vlevo.

Stejně jako u pláten a většiny prvků stránky používáme i u navigací jemně zaoblené rohy, takže plně využíváme vlastnosti `border-radius` a stejně tak přechodů barev gradientů. Ve stylu většiny prvků stránky se vyskytuje gradient přecházející z šedé barvy do černé pro vyvolání kovového dojmu.

```
#nav1{
    text-align: left;
    width: 170px;
    padding-left: 15px;
    padding-right: 10px;
    border: solid 2px white;
    border-radius: 5px;
    margin-left: 570px;
    padding-bottom: 15px;
    background: linear-gradient(gray,darkgray);
    box-shadow: 5px 5px 5px black;
}
#nav2{
    position: absolute;
    text-align: left;
    width: 170px;
    padding-left: 15px;
    padding-right: 10px;
    border: solid 2px white;
    border-radius: 5px;
    margin-left: 780px;
    margin-top: -975px;
    padding-bottom: 20px;
    background: linear-gradient(gray,darkgray);
    box-shadow: 5px 5px 5px black;
}
```

Z kódu je dobře vidět, že stylování obou navigací je takřka totožné, liší se hlavně hodnotou `margin`. Jelikož je nutné, aby oba výběry byly vedle sebe, tak musí být každý z nich pozicován odlišně.

5.1.6. Select

Další důležitou součástí stránky jsou tagy `<select>`, které jsou společně s `canvas`em nejdůležitějšími prvky stránky a pomocí nich probíhá veškerá volba jak vizuálních, tak i nevizuálních prvků. Všechny roletky `select`ů mají společné formátování.


```

select {
    border: solid 3px white;
    background-color: grey;
    border-radius: 5px;
    color: white;
    font-size: 12px;
    width: 170px;
    height: 20px;
}

```

5.1.7. Button

Tlačítka máme na naší stránce celkem čtyři. Tlačítko vyčistit builder, vytisknout kytaru, uložit kytaru a objednat. Všechna tlačítka jsou nastýlována nejen pomocí svého typu, ale taky pomocí divu s názvem btns. Každé tlačítko je obaleno tímto divem. Tlačítko pro objednávku btn4 je jediné, které má ještě samostatné stylování pro změnu barvy. Byla pro něj vybrána červená, aby bylo na první pohled zřetelné a markantní.

```

#btns{
    position: inherit;
    bottom: 308px;
    text-align: left;
    width: 195px;
    height: 40px;
    border: solid 2px white;
    border-radius: 5px;
    margin-left: 780px;
    margin-top: 10px;
    background: linear-gradient(gray,darkgray);
    box-shadow: 5px 5px 5px black;
}
button{
    background: linear-gradient(gray,darkgray);
    width: 195px;
    height: 40px;
    color: black;
    font-size: 16px;
    font-weight: bold;
    text-shadow: 1px 1px black;
}
#btn4{
    color: darkred;
}

```

5.1.8. Box a FinalPrice

Box a FinalPrice jsou dva divy v našem HTML kódu, které stojí také za zmínku, protože oba plní velmi důležitou funkci.

Div box slouží k obalení všech prvků těla kromě záhlaví a zápatí a k jejich následnému zarovnání na střed celé stránky. Celý tento „box“ je vyplněn pozadím s texturou dřeva, která výborně kontrastuje s kovovým zbytkem layoutu a přináší oživení v podobě přírodního vzhledového prvku, který určitě s hudebními nástroji, jakožto s výrobky ze dřeva souvisí.

FinalPrice je div, který slouží jako prostor pro vyobrazení ceny, která se generuje volbou prvků při stavbě kytary. Proto je nastaven tak, aby zobrazoval text bílou barvou a kontrastoval tak s tlačítky, které jsou okolo. Má i výrazně vyšší velikost písma 20px, opět pro lepší zvýraznění.

```
#box{
    position: relative;
    background-image: url(img/coco.jpg);
    width: 1000px;
    height: 975px;
    margin: 0 auto;
}

#finalPrice{
    position: inherit;
    bottom: 308px;
    text-align: center;
    width: 195px;
    height: 50px;
    border: solid 2px white;
    border-radius: 5px;
    margin-left: 780px;
    margin-top: 10px;
    padding-top: 30px;
    background: linear-gradient(gray, darkgray);
    box-shadow: 5px 5px 5px black;
    color: white;
    font-size: 20px;
    font-weight: bold;
    text-shadow: 2px 2px black;
}
```



Obrázek 5.1. Náhled na výsledný layout

5.2. Realizace vykreslení kytary

Jak už bylo řečeno výše, k realizaci celého vykreslování je použito JavaScriptu. Konkrétně dnes už notoricky známé knihovny jQuery a dále našeho vlastního scriptového souboru main.js. Jednotlivé kroky vykreslení jsou řešeny za pomoci řady funkcí a podmínek, kterým se blíže budeme věnovat v této části práce. Také zde bude řečeno něco málo k tvorbě grafiky samotné.

5.2.1. Tvorba grafiky

Pro tvorbu grafiky byl použit software PhotoFiltre Studio X. Tento program umožňuje jednoduchou tvorbu výřezů, obrázků bez pozadí a standartní grafické úpravy jako změnu kontrastu, středních tónů atd.

Práce na grafice byla nemalou součástí celého procesu vytváření aplikace. Pro základní tři modely, které aplikace obsahuje, je vytvořeno celkem 83 různých PNG obrázků, které jsou použity při vykreslování jednotlivých komponent. V budoucnu bude toto číslo ještě mnohem větší díky rostoucímu množství možností a modelů, které budou firmou vyráběny.

5.2.2. Logika vykreslování

V původním návrhu bylo na logiku celého skládání prvků nahlíženo způsobem, při kterém se prvky nejdříve logicky poskládají, tzn. vyřeší se jednotlivé podmínky a seřazení podmínek vykreslování bude korespondovat se vzájemným překrýváním prvků. Takže jsme nejprve chtěli získat jednotlivé zvolené možnosti ze selectů a následně je ve správném pořadí vykreslit na jedno jediné plátno.

Tento způsob se ovšem v praxi neosvědčil, jelikož implementace prvku canvas v prohlížečích není dokonalá. Ač bylo vykreslování logicky seřazené, většina prohlížečů stejně vykreslování prvků prováděla velmi nedeterministicky. Takže docházelo k situacím, kdy se například při změně barvy těla kytary v pozdějších fázích stavby stalo, že tělo překreslovalo prvky, které byly logicky na něm, a nebylo je tedy možno vidět.

Tento problém byl nejhorší v prohlížeči Google Chrome, kde docházelo k absolutně náhodnému vykreslení prvků. Jednou se kytara vykreslila dobře a pak třikrát úplně špatně. Velmi podobné to bylo také v prohlížeči Opera.

Nejlépe si s tím poradil prohlížeč Mozilla Firefox, který prvky zhruba v 90% případů, skládal ve správném pořadí, ale chyby se občas také objevovaly.

Toto byl pro aplikaci celkem zásadní problém, proto jsme museli vymyslet alternativní řešení. Po zvážení možností a promyšlení celé situace se zdála jako nejlepší a nejjednodušší možnost použití několika přes sebe ležících transparentních prvků <canvas>, z nichž každý má stejné rozměry, díky transparentnosti je v podstatě neviditelný a má svůj vlastní z-index, který deklaruje jeho pořadí v naší „hromadě“ překládajících se vrstev.

Každá součást kytary má své vlastní plátno, na kterém se vykresluje a jednotlivé prvky jsou samozřejmě logicky přiřazeny k plátnům tak, aby prvek, který je na vrcholu hierarchie zobrazování, byl přiřazen k plátnu s nejvyšším z-indexem.

Máme zde také jedno dodatečné plátno, které neslouží k zobrazování žádného prvku při stavbě, ale slouží jako „spojovací“ plátno, na kterém se všechny prvky překreslí do jednotného obrázku formátu PNG. Toto je čistě kvůli funkci tisku a funkci uložení obrázku.

5.2.3. JavaScript pro vykreslení

Nejen vykreslování, ale celá aplikace je založena na jedné velké funkci GuitarBuilder v main.js, která obsahuje ve svém těle všechny nutné dílčí funkce. Dále zde máme globální proměnnou myGB, která je vlastně instancí funkce GuitarBuilder a po celou dobu jsou v ní uchovávána veškerá vypočtená data a informace o jednotlivých krocích.

```
var myGB = new GuitarBuilder();
```

Ve výše zobrazeném řádku kódu můžeme vidět tvorbu instance GuitarBuilder, která je potom využívána k veškerým procesům volaným pomocí jQuery výrazů.

Samotné vyobrazení prvku na plátně je realizováno funkcí canvasDraw, která slouží k získání 2D kontextu všech dostupných pláten, které máme v našem HTML kódu a následně vytvoří novou proměnnou imageObj typu Image. Nakonec zavolá na daný imageObj funkci onload, která obrázek vykreslí do příslušného plátna.

```
this.canvasDraw = function(canvas_id, imgPath, x, y) {  
    var ctx = $("#myCanvas" + canvas_id)[0].getContext('2d');  
    var imageObj = new Image();  
    imageObj.onload = function() {  
        ctx.drawImage(imageObj, x, y);  
    };  
    imageObj.src = imgPath;  
};
```

Parametry funkce canvasDraw jsou canvas_id, což je v podstatě číslo přidáné nakonec id hodnoty každého plátna, dále imgPath, což je cesta k obrázku a nakonec souřadnice x a y, které jsou využívány pro umístění obrázku na plátně.

Vše výše zmíněné se děje na úrovni pláten a jednotlivých case případů. Celkové překreslení je realizováno pomocí funkce drawPreview, která zastřešuje všechny switch klauzule pro vykreslení a při zavolání způsobí překreslení všech zvolených prvků pomocí canvasDraw.

5.2.4. Podmínky vykreslení

Podmínky vykreslení jednotlivých prvků jsou dost individuální podle prvku samotného, ale obecně jsou v našem projektu řešeny pomocí klauzulí switch a následnými případy case:, ve kterých můžou nebo nemusí být použity různé podmínky typu if{}, else if{}, else{}. Ukážeme si celou strukturu v praxi na pár příkladech.

```
switch (this.gtr)
{
    case "no":
        break;
    case "lp":
        if (this.color === "no") {
            this.canvasDraw(1, 'img/lp.png', 150, 400);
        } else {
            this.canvasDraw(1, 'img/lp_color/lp_' + this.color + '.png',
150, 400);
        }
        break;
    case "sg":
        if (this.color === "no") {
            this.canvasDraw(1, 'img/sg.png', 147, 410);
        } else {
            this.canvasDraw(1, 'img/sg_color/' + this.color + '.png',
147, 410);
        }
        break;
    case "fly":
        if (this.color === "no") {
            this.canvasDraw(1, 'img/fly.png', 108, 415);
        } else {
            this.canvasDraw(1, 'img/fly_color/fly_' + this.color + '.png',
108, 415);
        }
        break;
}
```

Zde můžeme vidět vykreslování těla kytary, jehož id je gtr, ve kterém je použita podmínka if následujícím způsobem. Pokud není v select boxu color vybrána žádná barva, tzn. je nastaven na hodnotu „no“, pak vykreslí funkce canvasDraw výchozí obrázek tvaru těla, což je obrázek s vysokou průhledností, který má jen nastínit tvar těla. Ovšem pokud je

v select boxu color zvoleno nějaké barevné provedení a hodnota se tedy nerovná „no“, tak se vykreslí přímo daný typ barevného provedení.

```
switch (this.tuner)
{
    case "no":
        break;
    case "gold33":
        if (this.head === "lp1") {
            this.canvasDraw(9, 'img/tuners/lpgold.png', 212, 60);
        }
        else if (this.head === "lp2") {
            this.canvasDraw(9, 'img/tuners/lpgold.png', 212, 60);
        }
        else if (this.head === "ec") {
            this.canvasDraw(9, 'img/tuners/lpgold.png', 212, 60);
        }

        else {
            this.canvasDraw(9, 'img/tuners/fvgold.png', 209, 60);
        }
        break;
}
```

Zde vidíme podobnou situaci při vykreslování ladících mechanik, které mají id tuner. Jejich pozice je totiž závislá na zvoleném tvaru kytarové hlavy, takže při zvolení určitého typu hlavy v selectboxu s id head je nutné, aby k němu byly vykresleny příslušné ladící mechaniky.

Hlavy typu Flying V mají totiž mechaniky umístěny do tvaru „šipky“, zatímco ostatní tvary hlavy, které firma nabízí, mají mechaniky umístěny paralelně.

Toto je vyřešeno pomocí podmínek, ve kterých zjišťujeme, jaký typ hlavy byl zvolen podle jeho hodnoty value a následně přiřadíme příslušný obrázek ladících kolíků. To vše se samozřejmě musí udělat pro každé ze tří barevných provedení ladících kolíků.

```
switch (this.pick) {
    case "no":
        break;
    case "gold":
        this.canvasDraw(7, 'img/pickups/gold.png', 222, 450);
        break;
    case "chrome":
        this.canvasDraw(7, 'img/pickups/chrome.png', 222, 450);
        break;
    case "emg":
        this.canvasDraw(7, 'img/pickups/black.png', 222, 450);
        break;
}
```

```

        case "gold1":
            this.canvasDraw(7, 'img/pickups/gold1.png', 222, 520);
            break;
        case "chromel":
            this.canvasDraw(7, 'img/pickups/chromel.png', 222, 520);
            break;
        case "emg1":
            this.canvasDraw(7, 'img/pickups/black1.png', 222, 520);
            break;
    }

```

Zde můžeme vidět poslední ze serie „kreslicích“ switch klauzulí, které si ukážeme. Tato je na rozdíl od minulých dvou podstatně jednodušší. Jedná se o vykreslování kytarových snímačů, které je pro všechny nabízené modely naprosto totožné. Není tedy třeba jakýchkoli změn umístění pomocí souřadnic x a y.

Jediné, co je třeba rozdělit do samostatných případů, jsou jednotlivá barevná provedení snímačů a odkaz na obrázek, který je požadován. Vzniká nám tedy 8 různých případů vykreslení plus devátý případ, kdy je zvolena výchozí hodnota.

5.2.5. Počítání ceny

Podobně jako má vykreslování funkci drawPreview, tak počítání ceny má rovněž svou vlastní funkci nazvanou getPrice. Tato funkce si nejprve úplně na začátku vytvoří soubor proměnných. Pro každý volitelný prvek se vytvoří jedna proměnná a následně vynuluje. Poté využívá úplně stejných klauzulí switch jako funkce pro kreslení prvků s tím rozdílem, že nic nekreslí, ale podle zvolených případů přiřadí proměnné cena určitou hodnotu. V praxi tedy nejdříve vytvoří např. proměnnou `var priceGtr = 0;` a následně této proměnné přiřadí určitou hodnotu podle toho, k jakému případu v klauzuli switch došlo.

```

switch (this.board)
{
    case "no":
        priceNeck = 0;
        break;
    case "rosewood":
        priceNeck = 500;
        break;
    case "ebony":
        priceNeck = 800;
        break;
    case "maple":
        priceNeck = 500;
        break;
}

```


V tomto konkrétním kódu se jedná o nastavení ceny hmatníku kytary podle zvolené možnosti. Pokud není vybrán žádný typ dřeva cena hmatníku je nulová. Následně se mění podle zvoleného typu hmatníku.

Obdobně to funguje u všech prvků kytary. Cena je při změně jakéhokoli selectu automaticky znovu přepočítávána, takže zákazník neustále dostává informaci o aktuálním stavu celkové ceny jím zvolené kombinace prvků.

Počítání celkové ceny je obsaženo na samotném konci funkce `getPrice` a jedná se o prostý součet hodnot všech proměnných pro jednotlivé části. Nejprve se provede propočet, uloží se do parametru `computedPrice`, který je deklarován na samotném začátku funkce `GuitarBuilder` a následně je vrácena cena. Vracená cena se pak používá k zobrazení ceny v našem HTML souboru, konkrétně v divu `finalPrice`. Nyní si ukážeme, jak výše popsaný systém vypadá v kódu.

```
this.computedPrice =  
    priceGtr + priceBody + priceColor + priceBind + priceNeck +  
    priceHead + priceInlay + pricePickups + priceBridge +  
    priceTuners + pricePots + priceSwitch + pricePickguard +  
    priceBwood + priceNwood + priceNscale + priceFsize +  
    priceCase;  
return this.computedPrice;
```

V tomto kódu můžeme vidět samotný propočet ceny

```
$("#finalPrice").html(myGB.getPrice() + ", - CZK");
```

Tato část kódu zobrazuje jQuery příkaz, který slouží k vypsání aktuální výše ceny do divu s id `finalPrice` v našem HTML souboru.

5.2.6. Vyčištění zvolených možností

Nastavení všech hodnot na základní hodnotu je jednou z dalších důležitých funkcí naší aplikace. Jelikož může velmi často nastat situace, kdy si bude chtít zákazník vybrat úplně jiné možnosti kytary a nejjednodušším způsobem jak předešlou konfiguraci zrušit je nastavit všem selectům výchozí hodnoty a vyčistit všechna plátna.

K tomuto účelu nám slouží funkce `cleanUp` a `canvasClear`. Obě tyto funkce jsou volány tlačítkem `btn1` s popisem vyčistit builder.

```

this.canvasClear = function() {
    for (c = 1; c < 14; c++) {
        var canvas = $("#myCanvas" + c)[0];
        var ctx = $("#myCanvas" + c)[0].getContext('2d'); //context pro canvas
myCanvas
        ctx.clearRect(0, 0, canvas.width, canvas.height);
    }
};

this.cleanUp = function() {
    this.gtr = 'no';
    this.color = 'no';
    this.board = 'no';
    this.bind = 'no';
    this.head = 'no';
    this.pick = 'no';
    this.tuner = 'no';
    this.pot = 'no';
    this.hw = 'no';
    this.pswitch = 'no';
    this.pickg = 'no';
    this.hcase = 'no';
    this.bwood = 'alder';
    this.nwood = 'nmaple';
    this.nscale = 'gibson';
    this.fsize = 'j1';
    this.inlay = 'no';
    this.nradius = '10';
    this.nprofile = 'C';
    this.computedPrice = 0;
    this.updateControls();
};

```

Zde můžeme vidět deklaraci obou výše zmíněných funkcí canvasClear a funkce cleanUp. Stojí také za povšimnutí, že funkce cleanUp ve svém těle volá funkci updateControls, která slouží v celé aplikaci pro aktualizaci možností v jednotlivých roletkách. Takže nejdříve deklarujeme hodnoty jednotlivých selectů a následně se volá funkce updateControls, která jim příslušné hodnoty nastaví.

Při stisku tlačítka btn1 vyčistit builder se tedy postupně volají tři funkce cleanUp, canvasClear a nakonec opětovně funkce pro získání ceny, která je v tuto chvíli samozřejmě nulová.

```
$("#btn1").click(function() {  
    myGB.cleanUp();  
    myGB.canvasClear();  
    $("#finalPrice").html(myGB.getPrice() + ", - CZK");  
});
```

Zde můžeme vidět kód pro nastavení všech událostí, které se stanou po kliku na čistící tlačítko btn1.

5.2.7. Tisk kytary

Vytištění kytary je jedna z dodatečných možností nabízená zákazníkovi pro případ, že si svůj vytvořený nástroj budou chtít uložit na památku, ukázat svým přátelům a podobně. Tisk je aktivován při stisku tlačítka btn2 vytisknout kytaru. Pro tisky jsou využity tři různé funkce. Jedná se o funkce canvasMerge, canvasPrint, canvasClear13.

Jelikož je celá naše aplikace realizována jako soubor canvas pláten a každý prvek se vyskytuje vždy jen na jednom z nich, byl by tisk jako takový nemožný. Proto jsme přidali do aplikace speciální třinácté plátno myCanvas13, které slouží pouze k vyobrazení celé konfigurace na jednom plátnu.

Funguje to tak, že se nejdříve zavolá funkce canvasMerge, která načte obsah všech třinácti prvků canvas a následně se postupně překreslí všechna plátna do plátna myCanvas13. Tím je získán jeden celistvý obraz všech zvolených prvků v jednom plátně.

Následně je volána funkce canvasPrint, která načte obsah třináctého plátna a převede ho do grafického formátu PNG. Poté je tento obrázek otevřen v novém tiskovém okně prohlížeče a prohlížeč je na toto okno přesměrován. Zákazník pak může jednoduše vytisknout obrázek pomocí tiskových možností a nastavení tiskárny, které se mu automaticky objeví.

Úplně ve finále celého tisku je volána funkce canvasClear13, která slouží k vyčištění pouze třináctého plátna. Popis jednotlivých funkcí si ukážeme v následujícím kódu.

```
this.canvasMerge = function() {  
    var canvasMain = document.getElementById('myCanvas13');
```

```

        var ctx = canvasMain.getContext('2d');
        for (c = 1; c < 13; c++) {
            var canvas = $("#myCanvas" + c)[0];
            ctx.drawImage(canvas, 0, 0);
        }
    };

    this.canvasPrint = function() {
        var canvas = document.getElementById('myCanvas13');
        var d = canvas.toDataURL("image/png");
        var w = window.open('about:blank', 'guitar');
        w.document.write("<img src='" + d + "' alt='guitar'/>");
        w.document.close();
        w.focus();
        w.print();
        w.close();
    };

    this.canvasClear13 = function() {
        var canvas = document.getElementById('myCanvas13');
        var context = canvas.getContext('2d');
        context.clearRect(0, 0, canvas.width, canvas.height);
    };

```

Ve vyobrazeném kódu můžeme vidět popis a deklaraci všech tří výše zmíněných metod canvasMerge, canvasPrint a canvasClear13. Všechny tyto funkce jsou volány při kliku na tlačítko s id btn2 úplně stejným způsobem jako jsou volány funkce u předchozích tlačítek, takže není třeba samotné volání znovu zmiňovat.

5.2.8. Uložení obrázku kytary

Kytaru si můžeme uložit pomocí kliku na tlačítko s id btn3 uložit kytaru. Princip je úplně stejný jako u tisku s tím rozdílem, že se nevolá funkce canvasPrint, ale funkce canvasSave.

Takže nejdříve se zavolá funkce canvasMerge, poté se volá funkce canvasSave, která je v podstatě totožná s funkcí canvasPrint. Jediným rozdílem je, že nevytváří okno s tiskovými možnostmi, ale obyčejný prázdný panel, ve kterém se zobrazí obrázek kytary. Pomocí pravého kliku na myš a volbou „uložit obrázek jako...“ si pak zákazník může uložit obrázek kytary do svého počítače a dále ho využívat ke svým účelům. Nakonec se opět zavolá funkce canvasClear13 pro vyčištění třináctého plátna.

5.2.9. Objednávka pomocí JotForm

Objednávka byla další problematickou částí celé aplikace a její realizace nebyla jednoduchá. Firma si přála, aby jednotlivé objednávky byly přijímány pomocí e-mailu a aby každá objednávka kromě klasických údajů jako jméno, příjmení, adresa, telefon, e-mail atd. také obsahovala výslednou konfiguraci nástroje a samozřejmě také cenu této konfigurace.

Nejdříve bylo třeba vyřešit samotnou realizaci objednávky přes formulář. Místo tvorby vlastního systému a formuláře byl zvolen velmi jednoduchý a přesto efektivní způsob řešení. Využili jsme internetových stránek společnosti JotForm www.jotform.com, která nabízí velmi elegantní řešení jakýchkoli formulářů.

Na jejich stránkách je možno zdarma za několik minut vytvořit velice efektivní formulář obsahující všechny nezbytné prvky. To vše za pomoci pár kliků bez jakékoli nutnosti cokoli programovat. Všechny tradiční možnosti jako jméno, příjmení, adresa, země pobytu atd. jsou v JotForm implementovány a stačí jen vybrat ty, které budou ve formuláři potřeba. Celý formulář nakonec stačí propojit s určitou e-mailovou adresou, kam poté budou chodit všechny odpovědi. Jotformy mají v sobě také automaticky zařazeny kontrolní mechanismy například pro kontrolu správného zadání e-mailové adresy.

Na formulář je možno se odkazovat pomocí url adresy například stiskem tlačítka v aplikaci, což by bylo v mnoha případech dostačující v našem případě ovšem ne. Jelikož je potřeba do našeho formuláře dostat také informace o konfiguraci kytary. Firma JotForm naštěstí nabízí i možnost úplného stažení formuláře včetně jeho JavaScriptových knihoven a všech CSS souborů se stylováním. Nejlepší na tom všem je fakt, že i když je formulář stažen, všechna propojení se serverem JotForm jsou v něm stále nastavena, takže se každý vyplněný formulář stále odesílá na nastavený e-mail na www.jotform.com.

Poté co byl formulář vytvořen, stažen a zařazen do aplikace bylo třeba nějakým způsobem vyřešit zobrazení všech konfiguračních parametrů kytary ve formuláři. Z tohoto důvodu bylo ve formuláři vytvořeno textové okno Konfigurace Kytary, do kterého budou vždy při zavolání formuláře pomocí funkce `optionGetter` vloženy všechny parametry zvoleného nástroje v textové formě a nakonec i cena celkové konfigurace.

```
var w = window.open('form.html?data=' + data, 'blank', 'scrollbars= yes, toolbar =  
no,width = 500, height = 950');  
w.focus();
```

V kódu výše vidíme konec funkce optionGetter, která nejdříve načte textovou hodnotu všech zvolených možností v selectech, tu pak uloží do proměnné data a tuto proměnnou přidá jako parametr za url adresu našeho formuláře form.html. Tato funkce je volána při kliku na tlačítko s id btn4 objednat.

Po kliku se tedy otevře náš formulář vyžadující zadání jednotlivých údajů o zákazníkovi. V těle formuláře je pak vnořena JavaScriptová funkce, pomocí které formulář získá informace z prvku data, který byl přidán do url adresy. Tyto data následně funkce načte do textového pole Konfigurace Kytary.

```
<script type="text/javascript">
    document.getElementById("si" + "mple" + "_spc").value = "41057077468359-41057077468359";
    var data = decodeURIComponent(getUrlVars()["data"]);
    var re = new RegExp('<br>', 'g');
    document.getElementById('input_16').value = data.replace(re, "\r\n");

    function getUrlVars() {
        var vars = {};
        var parts = window.location.href.replace(/[?&]+([^\&]+)=([^\&]*)/gi,
function(m, key, value) {
        vars[key] = value;
        });
        return vars;
    }
</script>
```

Zde můžeme vidět funkci k získání dat o konfiguraci z url adresy. Jedná se o vůbec nejsložitější funkci v celé aplikaci, jelikož v ní byl použit takzvaný regulární výraz. Regulární výrazy se používají k vyhledávání v textu nebo k manipulaci s textem. Funkce s regulárním výrazem je použita konkrétně k vyhledání řetězce podle výrazu a nalezené výskyty nahradí řetězcem, který je zadán.

Po vyplnění všech nezbytných údajů a kliku na tlačítko odeslat se objednávka odešle na e-mail firmy. V e-mailu je objednávka jako příloha ve formátu PDF. Po zobrazení objednávka vypadá takto.

K.K. Guitars - Objednávka	
Submission Date	2014-04-17 16:14:49
Jméno / Příjmení	Jozef Novák
E-mail:	j.novak.33@seznam.cz
Telefon	() 735889673
Adresa	Ulice: Kajetána Tyla Město: Karviná Kraj: Moravskoslezský PSČ: 733 01 Země: Czech Republic
Poznámka	Dobrý den, chtěl bych si kytaru převzít osobně. Děkuji
Konfigurace Kytary	bodystyle: Les Paul (14000 CZK) color: black (1000 CZK) binding: 3ply celuloid (400 CZK) fingerboard: ebony (800 CZK) headstock: Les Paul Custom (150 CZK) inlay: abalony block (850 CZK) pickups: 2 gold passive humbuckers (2800 CZK) bridge: Gold Tune o' Matic (700 CZK) tuners: Gold (1200 CZK) potentiometers: 2x vol/2x tone black (200 CZK) switch: gold 3way selector (120 CZK) pickguard: black (500 CZK) body wood: mahogany (1000 CZK) neckwood: mahogany (500 CZK) neckscale: Gibson 24.75" neckprofile: D neckradius: 12" case: black (3000 CZK) price: 27270

Obrázek 5.1. Objednávka

5.3. Údržba a přidávání nových prvků

Ačkoli chce firma pokračovat v budoucí spolupráci, bylo by dobré zmínit se o základní údržbě aplikace a budoucím rozšíření. Aplikace by samozřejmě mohla být vylepšována i firmou samotnou. Stačí, aby některý ze zaměstnanců uměl základy práce s grafikou, HTML a JavaScriptem.

5.3.1. Rozšíření stávajících možností

Pokud by šlo pouze o rozšíření stávajících možností, jsou úpravy minimální. V případě přidání nového barevného provedení, nového typu snímačů atd., stačí pouze v nějakém grafickém editoru vytvořit danou grafiku v potřebných rozměrech a vložit ji do složky img. Následně přidat v HTML souboru aplikace novou možnost <option> do příslušného tagu <select>. Nové možnosti je třeba vytvořit atribut value a přidat textový obsah možnosti.

Poté co máme vytvořenou možnost v HTML souboru, stačí jen přidat nový case v algoritmu na počítání ceny a v algoritmu na vykreslení grafiky. Case musí mít hodnotu totožnou s atributem value, který má daná možnost v HTML souboru. Do cenového algoritmu je třeba přidat příslušnou hodnotu ceny za prvek a do vykreslení přidat odkaz na obrázek daného prvku.

5.3.2. Přidání nové možnosti

Přidání úplně nové komponenty je už poněkud složitější. Nejdříve je opět třeba přidat nový prvek do HTML souboru.

```
<select name="novy" id="new" class="gbNew updateView">
    <option value="no">-</option>
    <option value="val1"> text1</option>
    <option value="val2"> text2</option>
</select>
```

Hned na začátku je třeba se rozmyslet, jestli se jedná o prvek vizuální, či prvek nevizuální a podle toho daný <select> vložit do příslušné nabídky.

Poté je třeba přidat prvek do definic všech stěžejních funkcí v main.js.

```
this.new = 'no';
```

Nejdříve do základní definice prvků.

```
this.updateControls = function() {
    $(".gbNew").val(this.new);
};
this.cleanUp = function() {
    this.new = 'no';
};
```

Následně do funkce updateControls a cleanUp.

```
this.optionGetter = function() {
    var new =
document.getElementById('new').options[document.getElementById('new').selectedIndex
].text;
var data = "novy: " + new + "<br>"
};
```

Poté do metody optionGetter a do deklarace proměnné data dle výše popsaného způsobu.


```

this.getPrice = function() {
    var priceNew = 0;
    switch (this.new)
    {
        case "no":
            priceNew = 0;
            break;
        case "val1":
            priceNew = XXX;
            break;
        case "val2":
            priceNew = XXX;
            break;

    } this.computedPrice = ... + priceNew;
    return this.computedPrice;
};

```

Následně je nutné vytvořit a vynulovat novou proměnnou pro cenu prvku ve funkci `getPrice`. Poté vytvořit příslušnou `switch` klauzuli pro jednotlivé případy a přidělit jim ceny. Je také nezbytné nově vytvořenou proměnnou připočíst k proměnné `computedPrice`.

```

this.drawPreview = function() {
    switch (this.new) {
        case "no":
            break;
        case "val1":
            this.canvasDraw(CanvasID, 'img/new1.png', x, y);
            break;
        case "val2":
            this.canvasDraw(CanvasID, 'img/new2.png', x, y);
            break;
    }
};

```

V dalším kroku musíme stejný problém vyřešit ve funkci `drawPreview` pro vykreslení. Takže opět vytvořit novou klauzuli `switch` s příslušnými parametry a podmínkami.

```

$(".gbNew").change(function() {
    myGB.new = $(this).val();
});

```

Nakonec je třeba deklarovat nový jQuery výraz v úplně spodní části `main.js`, který bude obsahovat ukazatele na náš nový prvek. Tento výraz slouží k tomu, aby aplikace reagovala na výběr v `selectboxu`.

To jsou všechny kroky potřebné k přidání nového prvku do aplikace. Jediné co by za jistých okolností mohlo být ještě nutné přidat je nové plátno Canvas, ale to jen za předpokladu, že by nově přidaný prvek byl graficky nejvýš v hierarchii a překrýval by některé z nadefinovaných prvků. Jinak je možno využít již existujícího plátna, ale je třeba vhodně zvolit plátno tak, aby v kýžené oblasti nebyl žádný jiný prvek již vykreslován a aby plátno mělo dostatečně velký z-index.

6. Závěr

Cílem práce bylo vytvoření webové aplikace pro firmu vyrábějící zakázkové elektrické kytary, která postupně přechází na CNC výrobu několika modelů. Jednou z hlavních podmínek byla vysoká možnost nastavitelnosti prvků nástroje zákazníkem samotným, jednoduchost, přehlednost, grafické rozhraní, ve kterém bude sestavený nástroj v reálném čase vidět a zobrazení ceny dané konfigurace nástroje v reálném čase. Dále pak rozdělení aplikace na českou a anglickou jazykovou variantu a možnost odeslání okamžité objednávky, ve které bude obsažena aktuální konfigurace nástroje a jeho cena, na e-mail firmy.

K tvorbě celé aplikace jsme využili znalostí získaných během bakalářského studia. Nejvíce znalostí týkajících se tvorby webu, práce s HTML a stylování pomocí CSS stylů, dále pak také znalostí ze všeobecného programování. V neposlední řadě jsme rovněž využili řadu vlastních zkušeností s úpravou a editací grafiky, které jsme získali častou prací s grafikou a její tvorbou.

Během tvorby naší aplikace jsme se taky velmi důvěrně a poměrně do hloubky seznámili s programovacím jazykem JavaScript, který je dnes velmi hojně při tvorbě webu používán, s jeho možnostmi, strukturou, nejčastěji používanými knihovnami atd.

V teoretické části práce jsme se zabývali hlavně technologiemi, které budeme k tvorbě naší aplikace bezpodmínečně potřebovat, což byly převážně HTML5 a jeho prvek Canvas, stylování pomocí CSS3 a technologie JavaScriptu, která se v naší aplikaci stará o veškeré funkcionality.

V kapitole Analýza současného stavu webů konkurenčních výrobců kytar se věnujeme webovým aplikacím konkurenčních firem. Analyzovali jsme nespočet webů konkurenčních výrobců hudebních nástrojů, kteří podobné aplikace svým zákazníkům nabízí. Jelikož v česku podobné aplikace firmy zatím nemají, jedná se hlavně o zahraniční firmy používající podobný

typ aplikací jako je ta naše. Snažíme se vybrat to nejlepší na konkurenčních aplikacích a také se poučit z jejich nedostatků.

V poslední kapitole se zabýváme už konkrétním návrhem a následným řešením naší aplikace. Nejprve návrhem layoutu, poté návrhem zbytku vzhledu a návrhem vnitřní funkcionality naší aplikace. V části s řešením a implementací naší aplikace se věnujeme konkrétním funkcionalitám a principům, které jsme použili v naší aplikaci. Vyjadřujeme se k praktickému řešení vzhledových i programových prvků, k tvorbě grafiky prvků aplikace a v nemalé míře také k logice celé aplikace. Ve finální fázi se také vyjadřujeme k řešení objednávky pomocí formuláře JotForm.

Aplikace jako taková splňuje svůj účel velmi dobře. Grafické rozhraní funguje bez jakýchkoliv problémů a celé provedení bezpochyby zaujme nespočet potenciálních zákazníků. Firma chce pokračovat v další spolupráci rozšířením nabídky modelů a postupně i přidáním některých nových prvků, kterými plánuje v budoucnu své nástroje vybavit. Logika aplikace je napsána velmi přehledně a celková realizace aplikace umožňuje budoucí přidávání nových prvků. Jediné, co je třeba při přidání nového prvku vytvořit, je grafika. Co se kódu týče, je přidání prvků bezproblémové. Aplikace je samozřejmě dle požadavku dvojjazyčná. Celá aplikace je k vidění na adrese www.kkguitars.xf.cz/guitarbuilder.html. Byla zařazena do stávajícího webu firmy. Firma je tedy s aplikací spokojena a cíl bakalářské práce byl splněn v celém jejím rozsahu.

Nicméně realizace této aplikace nás přivedla k myšlence využití podobných aplikací nejen v prostředí výroby kytar a jiných hudebních nástrojů, ale ve svém důsledku by se podobný typ aplikací dal využít i v dalších odvětvích. Jedním z jasných kandidátů je třeba oděvnictví, kde už podobné aplikace dokonce existují. Zákazníci budou určitě rádi za možnost vyzkoušení různých kombinací outfitů, kterou mohou hned v reálném čase vidět a rozhodnout se tak například jestli tato kombinace oblečení dobře vypadá. Další v řadě je určitě také obuvnický průmysl. Například různé provedení součástí a vizuálních prvků bot, které si zákazník může poskládat dle libosti a vkusu. Jedním z mnoha dalších kandidátů na podobná rozhraní je například i cykloprůmysl. Představa jízdního kola, které si může zákazník sám upravit, by také stála za zvážení. Takže použití modifikovaných verzí naší aplikace by mohlo mít potenciál i do budoucna.

Seznam použité literatury

Tištěné knihy

GASSTON, Peter. *The Book of CSS3*. San Francisco: No Starch Press, 2011. 304 s. ISBN 978-1-59327-286-9.

GOLDSTEIN, A., L. LAZARIS a E. WEYL. *HTML5 a CSS3 pro webové designéry*. Přeložil RNDr. Jan POKORNÝ. Brno: Zoner Press, 2011. ISBN 978-80-7413-166-0.

HOGAN, Brian. *HTML5 a CSS3: Výukový kurz webového vývojáře*. 1. vyd. Brno: Computer Press, 2011. ISBN 978-80-251-3576-1.

KOSEK, Jiří. *HTML: tvorba dokonalých stránek podrobný průvodce*. 1. vyd. Praha : Grada, 1998. 296 s. ISBN 80-7169-608-0.

LIE, Håkon Wium a Bert BOS. *Cascading Style Sheets: Designing for the Web*. Addison-Wesley, 1997. 279 s. ISBN 978-02-0141-998-6.

PILGRIM, M., L. LAZARIS a E. WEYL, 2010. *HTML5: up and running*. Sebastopol, CA: O'Reilly. ISBN 05-968-0602-7.

Internetové zdroje

AGARWAL, Abhishek. *Differences between HTML and HTML5*. [online]. [cit. 2011-09-12]. Dostupné z: <http://www.rightwaysolution.com/blog/html-and-html5/>

BLAKE, Jason. *Infographics are a great way to learn! Infographics on HTML5 Cheat Sheets, Web Hosting Comparisons and SEO Audits*. [online]. [cit. 2014-01-05]. Dostupné z: <http://www.inmotionhosting.com/infographics>

KOSEK, Jiří. *Historie a vývoj HTML*. [online]. [cit. 2013-03-26]. Dostupné z: <http://htmlguru.cz/uvod-historie.html>

WOLFORD, Jack. *HTML5 Plugins 2013*. [online]. [cit. 2013-09-12]. Dostupné z: <http://themesquirrel.com/themeblog/html5-plugins-2013/>

Seznam zkratek

2D - dvourozměrný

3D – trojrozměrný

AJAX - Asynchronous JavaScript and XML

API - Application Programming Interface

CERN - Conseil Européen pour la recherche nucléaire

CNC - Computer Numeric Control

CSS - Cascading Style Sheets

HSL - Hue Saturation Lightness

HSLA - Hue Saturation Lightness Alpha

HTML - HyperText Markup Language

MIT - Massachusetts Institute of Technology

PDF - Portable Document Format

PNG - Portable Network Graphics

RGB – Red Green Blue

RGBA - Red Green Blue Alpha

URL - Uniform Resource Locator

WWW - World Wide Web

W3C - World Wide Web Consortium

XHTML - Extensible Hypertext Markup Language

XML - Extensible Markup Language

Prohlašuji, že

- jsem byl seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;

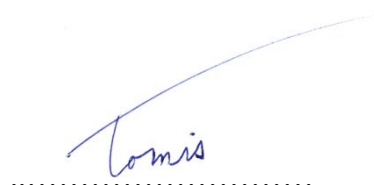
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, bakalářskou práci užít (§ 35 odst. 3);

- souhlasím s tím, že bakalářská práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího bakalářské práce. Souhlasím s tím, že bibliografické údaje o bakalářské práci budou zveřejněny v informačním systému VŠB-TUO;

- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;

- bylo sjednáno, že užít své dílo, bakalářskou práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne 7. května 2014



Tomáš Tomis

Adresa trvalého pobytu studenta:

Na Sedlácích 1011

739 34 Šenov